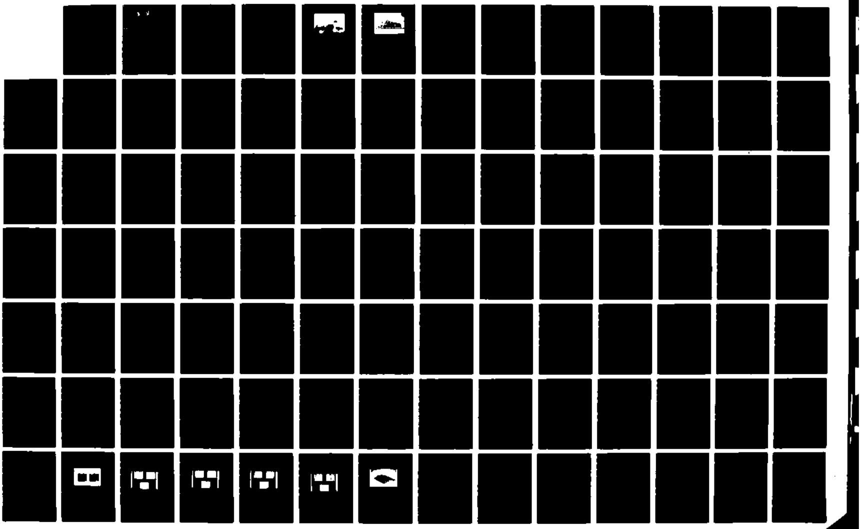
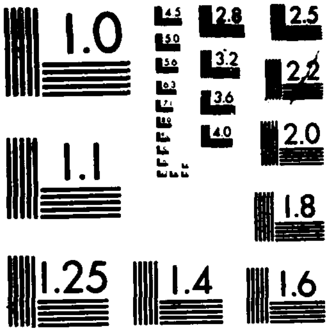
AD-A185 687   A QUERY DRIVEN COMPUTER VISION SYSTEM: A PARADIGM FOR   1/2
HIERARCHICAL CONTRO..(U) PENNSYLVANIA UNIV PHILADELPHIA
SCHOOL OF ENGINEERING AND APPL..   R BAJESKY APR 83
UNCLASSIFIED   AFOSR-TR-87-1205 F49620-83-K-0037   F/G 17/5   NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SECURITY CLASSIFICATION OF THIS PAGE

IENTATION PAGE

2

AD-A185 687

1a. REPORT SECURI
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLAS

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

DTIC FILE COPY

2b. DECLASSIFICATION/DOWNGRADI SCHE

OCT 1 3 1987

S D

4. PERFORMING ORGANIZATION REP MBER(S)

D

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 87-1205

6a. NAME OF PERFORMING ORGANIZATION
University of Pennsylvania

6b. OFFICE SYMBOL
(If applicable)

7a. NAME OF MONITORING ORGANIZATION
Air Force Office of Scientific Research

6c. ADDRESS (City, State and ZIP Code)
School of Engineering and Applied Science
University of Pennsylvania

7b. ADDRESS (City, State and ZIP Code) Bld 410
Directorate of Mathematical & Information
Sciences, Bolling AFB DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL
(If applicable)
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F49620-83-K-0037

8c. ADDRESS (City, State and ZIP Code)
Bld 410
Bolling AFB DC 20332-6448

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A1	

11. TITLE (Include Security Classification)
A Query Driven Computer System:A Paradigm for Hierachical control Stragties During the recognition process of three dimensional visually perceived objects

12. PERSONAL AUTHOR(S)
Bajesky

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 7/1/84 TO 12/31/86

14. DATE OF REPORT (Yr., Mo., Day)
April 1983

15. PAGE COUNT

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Two necessary components of any image understanding system are an object recognizer and a symbolic scene representation. The LandScan system currently being designed is a query driven scene analyzer in which the user's natural language queries will focus the analysis to pertinent regions of the scene. This is different than many image understanding systems which present a symbolic description of the entire scene regardless of what portions of that picture are actually of interest. In order to facilitate such a focussing strategy, the high level analysis which includes reasoning and recognition must proceed using a top-down flow of control, and the representation must reflect the current sector of interest. This paper proposes the design for a goal-oriented object recognizer and a dynamic scene representation for LandScan a system to analyze aerial photographs of urban scenes. The recognizer is an ATN in which the grammar describes sequences of primitives which define objects. The Scene Model is dynamically built as the objects specified by the queries are recognized. Thus the control of the scene modelling is top-down, reflecting the user's interest in the scene. The Scene Model represents both the objects in the image and primitive spatial relations between these objects.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Maj. James Crowley

22b. TELEPHONE NUMBER
(Include Area Code)
(202) 767-5025

22c. OFFICE SYMBOL
NM

EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AFOSR-TR- 87-1205

A Query Driven Computer Vision System: A Paradigm for Hierarchical Control Strategies During the Recognition Process of Three-Dimensional Visually Perceived Objects

1. Introduction

In our proposal "Query Driven Computer Vision System: A Paradigm for Hierarchical Control Strategies During the Recognition of Three-dimensional Visually Perceived Objects", written two years ago, we set out to build a system which is able to interpret a natural language query and automatically generate a recognition strategy. We listed as key features of the proposed system:

1. automatic generation of recognition strategies
2. natural language input and output
3. hardware implementation of hierarchical architecture for real time processing, including real time stereo computation.

Since it was a proposal for four years, we shall first describe our accomplishments during the last 18 months, and based on our experience and progress, outline what we wish to do the next two years. This research is a part of a larger research effort conducted in the GRASP (General Robotics and Active Sensory Perception) Laboratory, which in turn is a part of the Center for Artificial Intelligence at the University of Pennsylvania. The Center for AI is supported by two large five year grants: one coming from NSF-CER (Computer Experimental Research), which goes from September 1983 through August 1988, and the other coming from the Army Research Office, which goes from September 1984 through August 1989. The principal investigators on both of these grants are Professor A.K. Joshi with R. Bajcsy as Co-PI, and a few other Computer Science Professors making various contributions. All the equipment in the GRASP laboratory, except for the IKONAS image display (which was purchased from this Airforce Grant) has been purchased from these two large grants. Needless to say that due to the Center for AI and its funding, the research proposed in this grant is well backed in terms of facilities, (see also the section on Facilities) but we need the support for people in order to carry out the work.

We emphasize the role of the **active sensor** in our research. By active sensor we mean a camera(s) which can move and serve as a probe rather than just a static recorder of the scene. This should not be confused with active sensors like sonar, radar, structured light, and laser range finders, which actually transmit a signal into the environment and receive its echos. The human analogy for the active sensor paradigm is a pilot in an airplane who can move his/her head and eyes in order to improve the recovery of the 3D information by combining stereo with motion, improving the visibility of some details by control of zoom and focus, and their like. The activity is not in transmitting signals, but in positioning the sensor and optimizing its parameters for the signals being received.

The second area we emphasize is the Natural Language (NL) query. This is the reason why in this first phase we have concentrated more on the systems issues than on the perfect solution to individual modules. We wanted to provide a pictorial system (with depth map, surface descriptions, etc.) so that the Natural Language queries could be executed. Due to the query the user is continuously interacting with the system and the perceptual domain. The query represents the objects and their spatial relationships in the scene which must be translated into those components that the perceptual module can identify. This of course implies a study of modularity and specialization and yet interaction between the purely perceptual entities, and the linguistic properties.

The last but not least component of this research is the aspect of real time processing. Here we are interested in the analysis of established perceptual algorithms that can be converted into parallel algorithms, and in the development of high performance computer architecture for their implementation.

All this research though basic is also very experimental. Because of the complexity of the scenes, sensing apparatus, and the processing strategies, we are testing the system with both real life photos as well as on a scene mock-up, or model. This latter capability is provided by a controlled and verifiable experimental environment including arrangements of known objects to form the investigated scene. For this purpose we use two scale models: one of a general city scene (Figure 1) and another of the engineering quadrangle of the University of Pennsylvania in Philadelphia (Figure 2). The latter is scaled at 300:1 and the objects are quite detailed. The importance of the controlled scene is that we can test the "goodness" (including accuracy and precision) of our vision operators by making actual measurements of the objects and comparing them to the scale model. Furthermore, we can use these scenes as a testbed for comparative studies of our vision operators/algorithms with similar operators from other laboratories. The basic research issues that we have been concerned with all along in this program are as follows.

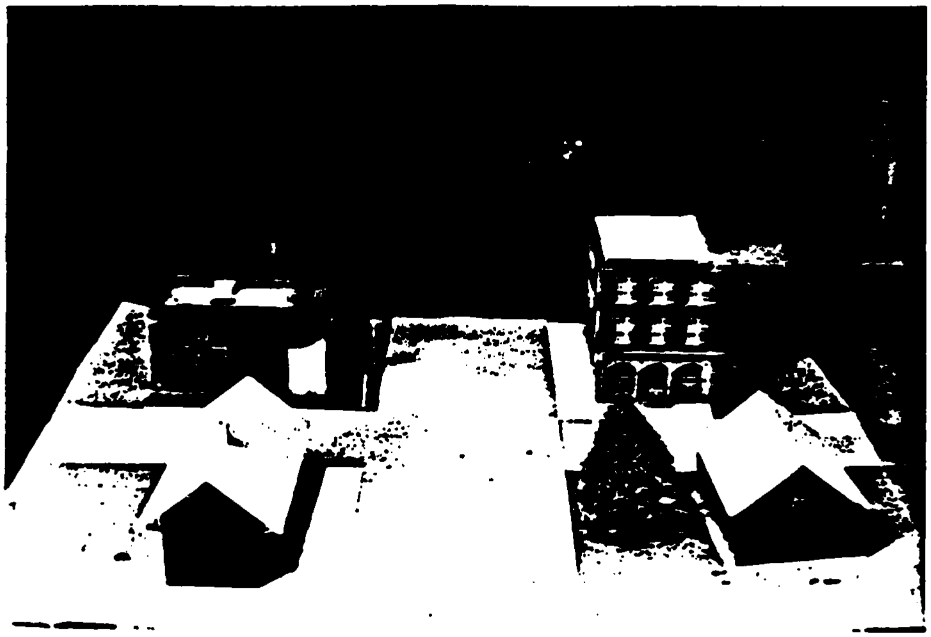


Figure 1-1: General City Scene

1.1. Computer Vision

1. On the low level image processing we are investigating the robustness and the uncertainties of the low level visual operators, like the edge detectors, under different illuminations, different orientation, focus and zoom of the cameras.

2. For the recovery of three-dimensional information we are interested in how to combine redundant information and resolve conflicting data, such as what comes from stereo and optical flow.

3. Rules for recognition strategies: Are there any principles? Can we separate the rules based on the knowledge about the camera parameters, the illumination and the semantics of the objects?

1.2. Natural Language

1. Since this is a query driven system, the user can employ NL words to specify the spatial relations between the objects in the perceptual domain. One of the research issues then is to develop a computational model which maps these linguistic terms onto the perceptual model of the scene. This model must account for the meaning of the words which are related by the locative construct (i.e. spatial construct).

2. Also due to the query the user is continuously interacting with the system and

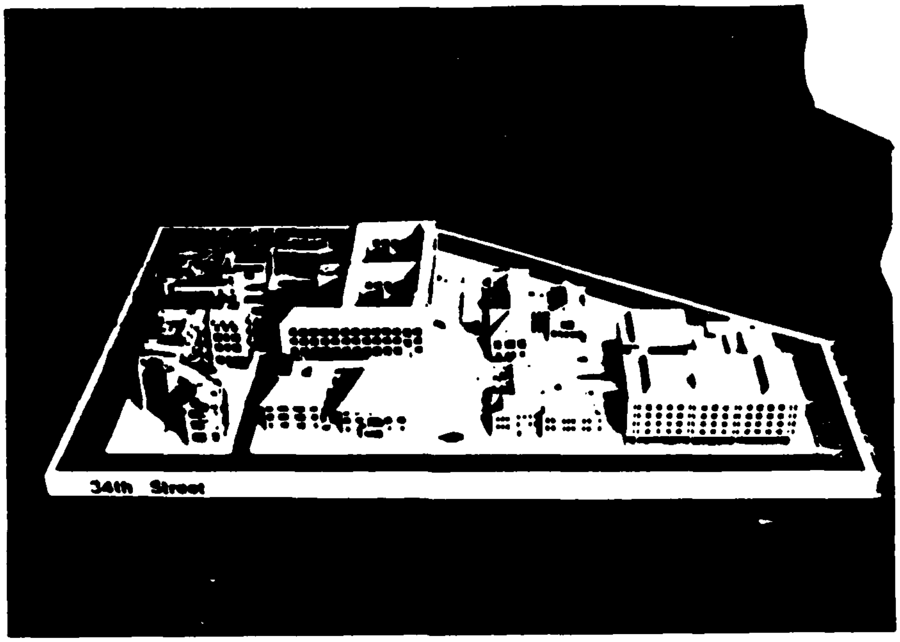


Figure 1-2: Engineering *Quadrangle of the University of Pennsylvania* in Philadelphia

perceptual domain. We intend that there will be some cooperative behaviour between the system and the user. Here, however, we have extra degrees of freedom stemming from the active sensors, and their probing of the environment, that adds to the dynamics of this particular system. Thus one of the fundamental goals of this research is development of a **computational model** that accommodates this kind of interaction.

3. Last but not least, the development of NL interfaces to an active perceptual module involves some key issues of knowledge representation, modularity, and communication between the linguistic/conceptual and perceptual components of the system.

1.3. Special Purpose Computer Architecture

1. We are investigating both hardware and software issues relating to the implementation of ultra-high performance systems for the execution of low and medium level image processing algorithms.

2. In terms of hardware, the Image Processing Optical Network or IPON is being developed as a high performance MIMD system based on a non-blocking optical interconnection network. A basic attribute of IPON will be the dynamically partitionable and reconfigurable network based on optical-hybrid technology for key components to provide high bandwidth communications, high capacity

buffering, and certain types of high speed processing.

3. User level programming of IPON will be accomplished using the concept of process level dataflow control via an interactive graphical image processing language. Of fundamental importance here is the design of optimal strategies for the static and dynamic allocation of resources (processors, memory, communications links) and real-time scheduling.

1.4. Outline

In the subsequent chapters we shall describe in more detail our results for the last year and a half and our plans for new research. It will be divided into three parts:

- the computer vision investigation,
- the natural language problem, and
- the special purpose architecture development.

2. Computer Vision

The computer vision section will be further subdivided into three sections:

- the low level image processing with active sensor
- the recovery of 3D information;
- and the surface reconstruction, representation and interpretation.

2.1. Low Level Image Processing with Active Sensors

Traditional approaches with static images use much low level image processing which concentrates on filtering and edge detection. In the context of active sensing we are seeking measurements from the current scene to feed back and control the various parameters of the active camera: size of the lens aperture, positioning of the head, orientation and the viewing angle, zooming in on the area of interest and converging on some points of interest with the vergence control of the stereo camera.

We have investigated several edge detectors and filters in the domains of both time and space. In particular, we have experimented with a non-directional edge detector very much like the Laplacian of Gaussian function, a directional edge detector using the Gabor filter, another directional edge detector approximating the first derivative of intensity [Canny84;David84] and features of the intensity functions, such as the first and second derivatives, very much like Haralick's Topographic Primal Sketch [Haralick83;Crowley].

It is very clear that different filters and features are suitable depending on the scene, its illumination and the opening and closing of the camera aperture (iris). The open issues are:

a) what is the feedback signal for the camera in terms of opening and closing the aperture with respect to the optimal contrast.

b) How should differently scaled filters and their corresponding edges be combined in order to obtain the "best boundaries" of objects. Here we define contours as 2D outlines that are obtained from edges, and the label boundary denotes the true 3D boundaries of objects.

For this we propose the following study: a laboratory set-up with a fixed scene, for example a mock-up of a fictitious city (see Figure 1) with a 10-channel illumination setup which can be precisely computer controlled. What one wishes to measure is a function of the magnitude of an edge with respect to changes of two parameters: first, the illumination of the scene, the size of the aperture; second, the scale (bandwidth, standard deviation) [Witkin83;Terzopolous82] of the filter which is used before the edge detector is applied.

We hope to prove or disprove two hypotheses: one, that for every scene (depending on the material of objects in the scene) and the illumination there is an optimal degree of opening of the camera's aperture; the other is that the scale on which the edge is detected the "best" is proportional to the size of the object and to the detail that the observer is interested in.

Other low level image processing consists of linear and non-linear filtering (see Appendix 2).

2.2. Recovery of the 3D data

In this section we wish to study how to recover the 3D information from a stereo pair of images, a series of images taken in time, and controlling the vergence angle between a stereo pair of cameras.

2.2.1. Stereo

The problem of stereo is traditionally divided into two parts: the correspondence problem (which is the difficult one), and computing the true (in some absolute coordinate system) depth value. We assume that the camera calibration problem has been solved, including the problem of scan line registration [Izaguirre85]. First we shall deal with the problem of correspondence and matching. The computation of the true depth value will be treated when we discuss the use of the vergence angle.

The stereo matching problem: During the last year or so we have experimented with a combined edge-region matcher (Appendix 1). Although the results were encouraging, we wished to understand the inherent limitations of a stereo matcher of static scenes. Hence, we embarked on the following problem: Given two 2-D projected views of a 3-D scene which differ by an arbitrary but known transformation, one needs to find unique matching between corresponding points. We assume that the input data for both images is a series of edge maps recovered through different filters and/or features.

There are two possible errors:

1. features in each image that should be matched but are not--the true negatives;
2. the features in each image that should not be matched but are matched--the false positives.

Furthermore, from the total number of features not all have a match, due to partial occlusions. So the total number of matchable features is less than the total number of features in either image.

What are the parameters or features upon which matching may occur?

1. edge points
2. edge segments
3. two edges and their relationship (corners, intersection,..)
4. more then two edges
5. enclosed contours.

In other words, the feature vector can be ordered with respect to the number of components.

The selection of the particular feature from the above list (and there could be more) depends on two criteria:

- *Uniqueness*, i.e., we wish to have such a feature that uniquely finds its corresponding match; and

- *Robustness*, i.e., we need such a feature which will not be sensitive to the camera transformation.

From the uniqueness condition it would appear that the feature should be as rich as possible (ideally the whole object). On the other hand, from the robustness condition would follow the requirement for as small feature as possible. Our task is to find the optimum compromise between the two extreme criteria.

Adapting the methodology from the optimal control theory [Bryson, Ho69] we can translate our problem as follows:

our decision parameters are: the number of feature points (denoted u's)

the state parameters are: the complexity of features translated into the length of the feature vector, (denoted x's)

the constraint relations are two: robustness and uniqueness related to each other by their reciprocity or complementarity.

Actually, the uniqueness function is equivalent to the complexity of features. Hence the constraint relation can be reduced to one linear function

$$f(x,u) = x + mu - c = 0,$$

where m, c are constants.

The performance index, which is a scalar function of both decision and the state parameters, is in our case the error of matching. In many physical systems and/or problems the performance index function can be derived as an analytic function. However, in our case this function has to be derived only empirically because it depends on the complexity of the scene, which is impossible to model in its full generality. The best we can say is that we hope that the function will be nonlinear with the existence of minima so that one can derive an optimal control strategy. For obtaining the error function we propose the following procedure:

1. We shall analyze several stereo pairs from different domains. Using first just the edge maps, we shall perform matching and compute the total of error of false positives and true negatives, normalized with respect to all the points that should have been matched.

2. We display these values against the total number of edge points.

3. We compute larger features and perform matches on them and display the error as defined in 1. against the total number of features.

4. We repeat this process on larger and larger features.

5. From the obtained data we shall fit a function error (x,u) and then compute the minimum value satisfying the constraint f(x,u)=0.

6. From the minimum we should obtain the optimal feature length u for a given scene. This feedback signal also can be used for control if the pan/tilt of a slightly different view of the scene is required.

2.2.2. Optical Flow

The problem: Given a series of images and a particular feature in time, the problem is to compute the vector (its magnitute and direction) of the feature spatially displaced over time. The problem is similar to stereo computation in that the issue is to find the proper features upon which one can match and then solve the correspondence problem. The problem is different from the stereo in that while in stereo there is an angular disparity, in the time sequence when sampling rate is high the positional disparity between the consécutive images is purely translational.

For the computing of optical flow we have investigated the following features:
No features--the Horn and Schunk method; [Horn85]
Motion energy--Adelson's method [Adelson84]
Burt's correlation method [Heeg].

The advantage of the first two methods is that there is no need for solving the correspondence problem. However, the price for that is high! In Horn and Schunk's method the smoothness constraint is a terribly limiting factor. In Adelson's method we are getting only the motion energy and the movement direction left and right, no other. This method uses filters sensitive to space/time oriented intensity changes This work is in progress and it still remains to be seen whether we will be able to use this method for recovery of 3D from motion parallax.

2.2.3. Focus

Three-dimensional data can also be recovered from a scene using "depth from focus". We are building hardware to automatically control focus. We are developing four different techniques for measuring focus sharpness, including (in increasing computational complexity) scan-line sum-modulus-difference of intensity, grey-level population entropy, grey-level variance, and power spectrum energy distribution analysis (via radial histogramming).

These techniques will be implemented and compared with respect to their effectiveness in improving focus to the extent that one point in the visual field can be said to be in focus, and from the position of that point on the image plane, the camera focal length, and the diameter of the aperture, we can precisely and uniquely determine the range of that point.

2.2.4. Vergence Angle

The last method in the recovery of 3D information is the use of vergence angle. This is a direct way of reading out the distance once the correspondence of the point has been established. The method is essentially triangulation. We are building hardware to both control and measure the vergence angle between two cameras. With this angle, the exact distance to any point fixated in both visual fields can be discovered. Given this exact distance, the relative depth maps returned from stereo and optical flow can now be fixed as absolute depth maps. See Appendix 1 ("From Disparity to Depth") for details of how absolute depth maps are generated in the present implementation.

We propose to use this device (designed and under construction) for accurate and unique absolute distance mapping of the visible surfaces and the stereo and the optical flow for filling in the gaps, which return *relative* distances.

2.3. Surface Reconstruction and Representation

From the previous section it should be clear that no matter how hard we shall work on various algorithms to obtain as perfect as possible 3D data, there is an inherent limit, due to well known physical limitations (occlusion, illumination, focus, zoom, orientation and the visible aspect of the object, to name a few) to the completeness with which 3D information can be recovered. So the next problem is how to supplement the missing data. The obvious answer is that some kind of interpolation method needs to be applied.

2.3.1. Depth Point Interpolation - Filling in the Gaps

The research issue for any scheme of filling the gaps is the trade-off between the measurements and the *a priori* information. We elaborate on this trade-off with an example. Let us suppose that we have a sparse array of 3D points after a stereo and/or optical flow computation. Remember we are left with some points that have not been matched either in the stereo matching nor in optical flow computation. In order to fill in the gaps we have several possibilities:

a) we can ignore the unmatched points, i.e., have confidence only in those points (measurements) that have been matched. Then assume, let us say, a linear (or any polynomial) model (the *a priori* information about the local surface). Based on this we perform linear (or polynomial) interpolation between the neighboring points.

b) An alternative to the case a) is instead of assuming the linear or polynomial models, which are inherently local, neighborhood models, assume a global smoothness constraint, which, using variational calculus, tries to fit the smallest and smoothest surface over the sparse data. [Grimson81].

c) The third possibility is to assume a local smoothness constraint in the depth values. Then reexamine the unmatched points (match them with the closest edgels in the other image) and check whether their depth value would satisfy the smoothness constraint with the neighboring points.

d) Finally, if, for example, from the outline we can identify measured objects, then clearly the "fill in gaps" process can use this information. An example of this case can be sidewalks or roads in aerial views.

As usual in machine perception, there is no one technique that works uniformly well in all cases. We believe that this is an integral part of the surface interpretation. One clearly needs all the above techniques available and then having a rule-based system use whichever give the "best" results. For example if we have one object in the view, then perhaps the third method is the "best". If one has reason to assume that one deals with objects that have only planar surfaces, then the first method might be adequate. The third method is the most versatile since it uses the most measurements and the least *a priori* information. The cost is in computation. We have implemented all of these, and some partial results are shown in [Smitley84] (Appendix 1).

2.3.2. Reconstructing and Representing Surfaces

Having a rich set of depth points available, the next problem is how to find closed boundaries, and from them, surfaces.

Finding boundaries of objects versus their surfaces are two complementary mechanisms which work simultaneously in a cooperative fashion. For the problem of boundaries there are two problems that we wish to differentiate: one is to find the boundary of an object in a complex scene, that is to singulate (or segment) an object; the other is to identify boundaries among surfaces in the same object. In the first case the problem is of a decomposition of the 3D visible space into individual objects, for example, by finding the smallest enclosing convex polyhedron. In the second case we are concerned with finding enclosed curves or connected segments of lines that enclose a continuous surface.

While the problem of singulation of an object is the Ph.D. thesis of E. Krotkov (see his proposal), in this paper we shall report on the program for finding boundary lines, also called wire frames. Naturally, we assume that all visible boundaries are true physical boundaries. The process starts with looking for points of high curvature and corners. From these points, a divide-and-conquer method of recursive decomposition finds that line which has the lowest curvature and shortest path. Another method for finding contours which instead of divide and conquer first generates all possible contours and then uses graph search for finding the "best" contour in terms of some cost function was investigated by Heeger [Heeger84]. This work, though interesting as a plausible computational model for the psychophysical phenomenon of subjective contours, is inefficient for practical implementation with current sequential hardware. For the future we need to improve our corner finder! (See Appendix 1 for a discussion of how edge detection may also directly identify "edgels" as corner features). After obtaining lines in between the corners and/or high curvature points we still need to know which of these contours are closed. The process that performs this task also creates a graph (a linked list of vertices, edges, and faces) which serves as the basic data structure for further, higher-level processing.

All the above procedures get leverage by virtue of the fact that our objects are polyhedral. What remains an open research question is how to proceed when the surfaces within boundaries are not planar. One method we shall investigate is converting the set of 3D points into two images, one representing the surface normals and the other the range information. Then by applying region growing and/or edge detection techniques one should be able to discriminate between planar and curved surfaces [Dane82]. The curvature of curved surfaces can be represented using splines [Allen84].

Returning to our simplified world, once we have the planar faces, then we can compute the surface attributes and relations. Here we are benefitting from the work of our colleague Professor Badler and his students at the University of Pennsylvania, who have developed a geometrical modeling system SurfsUP [Radack84]. In it, a face is defined by its enclosing 3D contours. Attribute values for each face in the surface graph are computed [Krotkov84]: compactness, centroid vector, (outward-pointing) normal vector, area, "type" (building, sidewalk, field, street, and unknown), and number of sides. These values are computed once and stored on an attribute list. Furthermore, topological relations such as above, adjacent (touching), contiguous (sharing an edge), contains (proper inclusion), looksadjacent, lookscontiguous (respectively adjacent and contiguous under perspective transformations). Relations (and indirectly their complements) are computed once and stored as Boolean arrays. Results of this stage are fed to the object interpreter.

2.4. Computer Vision Bibliography

[1] Adelson, Edward, H. and Bergen, James, R.
Spatio-Temporal Energy - Models for the Perception of Motion.
Computer Science Princeton N.J., RCA, David Sarnoff Research Center, October, 1984.

[2] Allen, Peter.
Surface Descriptions From Vision and Touch.
International Conference on Robotics (pp. 394-397), Atlanta, August, 1984.

[3] Bryson, E.E. Jr. and Ho, Yu-Chi.
Applied Optimal Control.
Blaisdell Publ. Co., 1969.

[4] Canny, John, F.,
Finding Edges and Lines in Images.
Computer Science MIT AI-TR-720, , 1984.

[5] Dane, Clayton Albert III.
An Object-Centered Three-Dimensional Model Builder.
Computer Science MS-CIS-82-50, University of Pennsylvania, 1982.

[6] Grimson, W.E.L.
From Images to Surface: A Computational Study of the Human Early Vision System.
MIT Press, Cambridge, 1981.

[7] Haralick, Robert, M., Layne, Watson T., Laffey, Thomas J.,
The Topographic Primal Sketch.
The International Journal of Robotics Research , 1983.

[8] Heeger, David.
Filling In The Gaps: A Computational Theory of Contour Generation.
Master's thesis, University of Pennsylvania, 1984.

[9] Horn, B.K. and Schunck, B.G.
Determining Optical Flow.
Artificial Intelligence Vol. 17:185-203, 1981.

[10] Izaguirre, Alberto, Pu, Pearl and Summers, John.
A New Development in Camera Calibration Calibrating a Pair of Mobile Cameras.
Computer Science GRASPLAB, Unviersity of Pennsylvania, March, 1985.
Proc. 2nd Int. IEEE Conf. on Robotics, St. Louis.

[11] Krotkov, Eric, P.
Construction of a Three Dimensional Surface Model.
Computer Science TR-MS-CIS-84-40, GraspLab 12, University of Pennsylvania, 1984.

[12] Radack, Gerry.
NASA Project Programmer's Guide.
Computer Science CIS TR-MS-CIS-84-02, University of Pennsylvania, 1984.

[13] Smitley, D., Goldwasser, S., and Lee, I.
IPON - Advanced Architecture for Image Processing.
In . 12th International Symposium on Computer Architecture, Boston, MA, June, 1985.

[14] Witkin, Andrew, P.
Scale-Space Filtering.
Eighth IJCAI (1019-1022), 8-12 August, 1983.
Karlsruhe, West Germany.

3. The Natural Language Issues

One of our major tasks is the development of a natural language (NL) query system interface to a visual (perceptual) system. The reason for using NL is not because we want to construct a cute interface, but rather because the use of NL provides "flexibility" to the user. There are many aspects of "flexibility" that make such faces attractive for conventional databases or knowledge bases, and, of course, these will carry over to the perceptual domain also. However, the particular aspects of "flexibility" that are directly relevant to our domain are as follows.

The user can employ NL terms (words) to specify the spatial relations (and later actions in the robotics domain) in the perceptual domain. It is in these terms the user can best characterize the domain. The system then has the responsibility to map successfully these terms on to the terms (or composites of them) to the perceptual module of the system.

The semantics of spatial relational words (eg. spatial prepositions) is extremely complex. Determining the proper interpretation of a spatial preposition is not merely a matter of matching a preposition with a single representation. The interpretation of spatial constructs depends heavily on the entities which are related by that construct [Herskovits84] [Talmy83], For this reason, the system will have available to it the linguistic properties of the objects which may appear in the domain as well as a set of interpretations for the location of constructs based upon the semantic values of the entities it relates. The linguistic properties are those features which affect the usage and interpretation of a spatial construct (phrases describing the spatial relations between objects). Since the domain is a visual one, each object in the domain will have a "place" associated with it. This is what Herskovits calls the canonical geometric description of a spatial entity (objects) [Herskovits84]. Ordinary solid objects (buildings, vehicles, people) are bounded closed surfaces. Geographical objects are entities with slightly imprecise boundaries - roads, rivers, and fields. Some other properties which must be represented are a prototype shape and the allowable deviations from it, the relative size, and characteristic orientation - i.e. a table stands on its legs normally. The typical geometric conceptualization will also affect the choice of spatial construct - is the object normally viewed as a point or line. Along with the typical geometric conceptualization is the typical physical context of an object. For instance, a door is normally viewed as begin in a wall. The normal function of an object, its relative size, it functionally silent parts and the actions commonly performed with an object will also be necessary for analyzing the spatial constructs.

For example, proper use of the preposition IN as in A is IN B involves not only computing containment (or partial containment) of A in B, but also assuring that B is in its normai orientation. Thus. in asking "Is the coin in the cup?" the user is assuming that the cup is in its normal orientation. If that is not the case and, say, the cup is upside down and the coin is under it, a response by the system "Yes" would be misleading, as it will tend to confirm the user's false presumption that the cup is in its normal orientation. An appropriate response is at least "No", but preferably (more cooperatively), "No, it is under the cup, the cup is upside down". Thus the system has to be sensitive to the normal orientation of objects in order to fully capture the semantics of IN.

The kind of cooperative behavior described above has been studied extensively in the context of NL interfaces to conventional databases or knowledgebases. Much of this theory and technology for these domains can be successfully carries over to the perceptual domain. However, NL spatial terms have not been systematically studied from the point of view of developing interfaces for perceptual domains. A rather preliminary study appears in [Herskovitz82]. However, this study is incomplete in may ways, especially in terms of the development of a computational model without which it is of no great value to our proposed task. Thus, one of our fundamental goals is the development of an appropriate computational model for the kind of interactional we want to support.

The second aspect of "flexibility" we call the query driven system. Given the number of relevant spatial relations between objects in a perceptual domain, it is impossible to precompute all the necessary relations. Our approach is "query driven" in the sense that, as a result of a query being asked, the system will compute the needed information from perceptual database as necessary. This dynamic behavior is not limited to just making some additional computations on already collected date, but will also involve acquiring new data, for example, by taking an additional view from a different angle (or getting new information from another modality), etc. The user is not constrained by what information has been collected already and what predicates have been precomputed. His queries will determine what information is needed to properly answer the query, and if that information is not available, then it will so inform the perceptual module. The perceptual module can then determine whether this new information can be computed from the data already gathered or whether it will require to get new data. Such behavior is initiated by the failure of the query at some level of interpretation. Such an opportunity is rarely available in the conventional databases, and even when it is available, it is of a very limited kind, as in the case of updatable databases.

If the reasoning processes fail to produce a positive response (the query fails to have an answer although it is syntactically correct), two types of query failure analysis are performed. The first type of query failure involves a query violating the global knowledge known about the domain. In this case, the system will respond with a message indicating that the query is conceptually ill-formed in this domain and why it is ill-formed. For instance, if the query asked how many walls the street had, the system would respond that streets do not have walls and that for that reason, the query is ill-formed. The other type of failure involves not finding the information requested in the scene model. In this case, rather than simply responding that the system was unable to find the data in question, of the scene with the old one in order to obtain a positive response to the query.

Thus the development of interfaces to an active perceptual module involves some key theoretical issues of knowledge representation, modularity, and communication between the linguistic/conceptual and perceptual components of the system.

3.1. The hypothesis generation and object recognition

The goal of the LandScan system is to perform query driven analysis for urban scenes. This places two constraints on the object recognition process: it must have top-down control structure, finding only those objects referenced in the query, and must encode global knowledge about a domain in which objects of the same type may have very different appearances. We have considered several different schemes for the representation of the global knowledge necessary to perform object recogniation such as frame based [Hwang83, Glicksman83], production systems, [Rosenthal81] and their like. We have finally settled for Augmented Transition Network (ATN) formalism because it enables the global knowledge to be encoded as a generative model for constructing objects from the primitives in the scene while driving the recognition in a top-down fashion [zwarico84].

The ATN formalism [Bates81], [Winograd83] was chosen to perform object recognition. Despite earlier failures using syntactic object recognition [fu82] we have found that a higher level syntactic approach works well in the urban environment. It appears that there are "rules" to describe the recognition of objects in the urban, aerial domain. These objects appear to be composed of planes in fairly regular fashion even though their appearances may be quite different. For example, while two buildings may appear quite different, the relations between the planes which comprise each may be the same. Earlier attempts at object recognition using a syntactic approach failed because the primitives which were combined were too low level (edges, etc), the matching sequences were too strict, and the domains were not appropriate for a syntactic approach. In LandScan, the primitives used are higher level (surfaces) and thus have more information associated with them. Unlike other syntactic pattern matching systems, the grammar rules in LandScan do not specify a strict matching sequence. Instead they specify the properties which must hold between the simpler components of an object. Since the rules are more general there are fewer in the system thus simplifying the recognition process. The grammar enables the global knowledge about object appearances to be encoded as a generative model for objects of indefinite appearances. This also differs from the Tropt and Walters ATN for 3-D object recognition [Tropf83] first generates an hypothesis and then uses the ATN to verify the hypothesis is correct. The ATN operates using a top-down control structure - enabling the object recognition to be a query-driven process. In LandScan the control structure used in recognition has been separated from the global knowledge used in the recognition process. Thus finding additional object types only involves adding syntactic rules for recognizing these objects. It also implies that the control strategy used can be changed as long as it can still use the grammar rules.

The Augmented Transition Network (ATN) is composed of three parts: the grammar, a dictionary, and an interpreter. The grammar represents the *a priori* or world knowledge that the system must have in order to recognize objects and assign "cultural" labels to subset of the scene. The dictionary presents the actual data which will be used in the recognition process- the surface model described above. The third component of the recognizer is the Lisp program which provides the control structure for the process. An object is recognized by traversing a network successfully.

The grammar as written is a two level network (this is considerably simpler than most ATN's which handle natural language utterances.) The bottom level concerns itself with the recognition of "simple objects." An object is simple if its further decomposition into parts will result in no entity which is in the domain of objects. For example, decomposing a building with a pitched roof will result in two halves of a pitched roof. Neither of these entities are considered objects in the domain - they are parts of objects. This level consists of the networks SIMPBUILD, SIMPSTREET, SIMPFIELD, and SIMPSIDEWALK. The top level combines the simple objects which were recognized in the first level of the network into "complex objects". A complex object is decomposable in a nontrivial way into at least one simple object. Each grammar rule represents the components and relations which must hold between those components in order to be considered an object or "sub-object". The components are specified by the arc type - either an object primitive (surface) or a simpler instance of the object. The tests associated with the arcs encode the relations which must hold between the components as well as providing further checking for component features.

As objects are recognized, a dynamic model of the scene is incrementally built by adding more information to it as further image analysis occurs. The scene model in 3-D MOSAIC [Herman83] is also incrementally derived as more data becomes available but the modelling process is data driven. LandScan builds a model using a query driven control. In other words, the modeller obtains more data as the user directs the vision system to analyze other areas of the scene which are of interest to him/her. Thus the Scene Model reflects the user's interest in the scene. The LandScan dynamic scene model is especially useful because it is flexible. the accuracy of the scene model increases as new data is acquired. Thus old hypotheses can be discovered false, deleted, and the scene model updated to reflect the more accurate understanding of the scene. In LandScan, when the scene analysis of a new image begins the scene model is empty. As questions are asked, the scene analyzer/constructor searches for the entities whose existence is in question using the object recognizer described above. As soon as the objects queried are found they are added to the Scene Model. Thus the Scene Model also reflects the history of the user's interest in the image. The dynamic scene model is composed of two components: a list of objects currently known to be in the scene and a set of matrices representing the p primitive relations hold between the objects on the object list. This design facilitates updating the scene model. To update the model the new object is simply added to the object list and the primitive relation matrices are expanded to include the relationship of the new object to all other objects in the model.

The first component of the scene model is the object list. The elements on this list are those objects which have been recognized during previous scene analysis operations.

These objects are represented only by polyhedral surfaces, conceptually the most primitive component of an object. Thus to the high level reasoner it appears that objects are composed of only bounded planes - primitives at one level of representation. The use of a single primitive at one level of representation. The use of a single primitive (or a set of primitives which are not composed from one another) is conceptually clean to work with and is adequate for modelling objects in this domain. Each instance of an object in the scene has the information associated with it which was determined necessary to facilitate further scene analysis. The components of an object record are a name, the list of faces (polyhedral surfaces) comprising the object, its location in Euclidean three space(average of the centroids of all the faces comprising the object), and a subtype which gives more specific information about the expectations one can have about the object.

The relations in the scene model represent the primitive relations or topological properties between objects in the scene. The relations are ADJACENT, CONTIGUOUS, LOOKSADJACENT, LOOKSCONTIGUOUS, ABOVE, and CONTAINS. They are defined over the set of all objects currently recognized in the scene. These relations are defined similarly to their counterparts in the Surface Model. The relations are represented by their adjacency matrices because the adjacency matrix is easily updated and makes composition of relations simple. The composition becomes a simple matter of boolean matrix multiplication for which there are many fast and efficient algorithms.

The combined use of the Scene Model and the object recognizer facilitates the following scene analysis operations: determining the relations, both complex and simple, among objects; locating and identifying specific objects and object parts. The existence of objects will be resolved in one of two ways - finding the object in the scene model by searching the object list, or using the recognizer to find a new instance of the object. To find an object part its face list will be searching until the part is found using the global knowledge about parts embodied in the object model. As for resolving the interpretation of locative constructs, the relations allow objects to be located relative to other objects in the scene using the matrix operations specified by the sematics of the spatial constructs. Suppose the question were asked, "Is there a car on the street?" An object of type CAR is ON an object of type STREET if the following primitive relations hold:

CONTAINS(STREET,CAR)
ABOVE(CAR,STREET)

The reasoner would determine if the CAR is ON the STREET by calculating the following relation composition:

CONTAINS * AVOE^T

which would be calculated by a simple matrix multiplication of the CONTAINS adjacency matrix and the transpose of the ABOVE adjacency matrix. So the understanding of relational expressions will be accomplished by composing the primitive relations.

3.2. Natural Language Bibliography

[1] Bates, Madeleine.
The Theoty and Practice of Augmented Transition Network Grammars.
In Leonard Bolc (editor), *Natural Language Communication with Computers.*
Springer-Verlag, 1981.

[2] Fu, K.S.
Syntactic Pattern Recognition and its Applications.
Prentice-Hall, 1982.

[3] Glicksman, Jay.
Using Multiple Information Sources in a Computational Vision System.
In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[4] Herman, Martin, Takeo Kanade, Shigeru Kuroe.
The3D MOSAIC Scene Understanding System.
In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[5] Herskovits, Annette.
Space and the Prepositions in English: Regularities and Irregularities in a Complex Domain.
1984.
Draft: University of California, Berkeley.

[6] Hwang, Vincent, Takashi Matsuyama, Larry Davis, Azriel Rosenfeld.
Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding.
Technical Report, Center for Automation Research, University of Maryland, October, 1983.

[7] Rosenthal, David.
An Inquiry Driven Vision System Based on Visual and Conceptual Hierarchies.
UMI Research Press, 1981.

[8] Talmy, Leonard.
How Language Structures Space.
Technical Report 4, Berkeley Cognitive Science Report, January, 1983.

[9] Tropf and Walters.
An ATN for 3-D Recognition of Solids in Single Images.
In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[10] Winograd, Terry.
Language as a Cognitive Process.
Addison-Wesley Publishing Co., 1983.

[11] Zwarico, Amy.
The Recognition and Representation of 3D Imagesfor a Natural Language Driven Scene Analyzer.
Technical Report, University of Pennsylvania, 1984.

4. IPON - Advanced Architectural Framework for Image Processing

This section outlines the organization and implementation of IPON in terms of both the hardware and programming environment, the progr:ss to date, and our future plans for this research effort. Additional details can be found in [Gold84] and [Smit84].

4.1. Introduction

One fundamental computational problem with image processing is the time needed to execute typical algorithms. This is especially severe with the types of image processing required for interactive image understanding applications. These algorithms deal with extraordinarily large quantities of data. A typical two dimensional image (512 x 512) consists of approximately a quarter megabyte of data. Voxel (3D) and time sequenced images consist of much greater amounts of data. Even the most powerful contemporary processors become ineffective when presented with such quantities of data. Many related applications such as a mobile robot trying to avoid obstacles as it moves require real-time processing capability (one image every thirtieth of a second). The use of ACTIVE SENSORs further increases this computational load since processing may need to be performed quickly at several different levels of detail or on slightly different data.

The objective of the IPON (Image Processing Optical Network) project is to investigate possible solutions to these problems. An architectural framework is evolving from this effort which is usable on current computation systems and will be directly applicable to emerging advanced technology as it becomes available in the future.

The realization of real time image processing has long been a goal of many researchers in computer architecture. Towards this end many different architectures have been developed. The applicability of MIMD, SIMD, pipelined and data flow processors have been investigated [Etch83] and each found to have the following types of problems:

1. Lack of flexibility (Pipelined and SIMD processors).
2. Complex awkward programming (MIMD).
3. Implementation Difficulties (MIMD and Data Flow).
4. Limited areas of efficient application (SIMD, Systolic array).

Image processing represents one of a class of computation applications which requires the manipulation of extremely large datasets. Traditional computer architecture including Von Neumann (SISD) machines as well as pipeline or systolic arrays, SIMD, and MIMD networks falls far short of the performance required for the real-time needs of machine perception, image analysis, certain types of image related computer graphics, object tracking, etc. Inherent in these approaches are bottlenecks associated with network communications and data storage.

4.2. Overview of IPON

The Image Processing Optical Network represents an architectural framework consisting of two major parts: the IPON hardware configuration and optical interconnection network and the integrated IPON software environment.

IPON is a computer system built around an optical interconnection network. Optical interconnection networks such as the one which we are designing provide solutions to many of the problems associated with the use of traditional electronic networks. Communicating through this network are a number (< 1000) of heterogeneous processors which need not be 'silicon' based.

The IPON programming environment facilitates the development and debugging of parallel image processing algorithms. The hardware and the software of IPON have been designed in such a way that programs written using the IPON program development system can be efficiently executed on the IPON hardware as well as on other multiprocessors or conventional superminicomputers.

It was essential to develop a system that is easy to program and debug while still providing parallel execution for increased throughput. The IPON hardware configuration represents a machine on which actual image processing algorithms will be implemented and used by vision and robotics researchers. Towards this end, IPON embodies the following, which make it a powerful system for developing real time image processing algorithms. IPON is a system of hardware built around an optical network which is:

1. Completely connected
2. Non-blocking
3. High speed
4. Dynamically reconfigurable
5. Expandable at a linear cost

These characteristics:

- Allow for maximum utilization of any number of ultra-high performance heterogeneous processors which can be easily integrated into the IPON system.

- Reduce the concern over the time taken to transmit data from one processor to another. This can reduce the difficulty of task scheduling since the transmission of data is not as costly as it is in traditional MIMD systems.

- Allow for the use of distributed control flow as opposed to a centralized token matcher or task dispatcher.

- Make IPON expandable. The network complexity increases linearly with the number of processors, not at the rate of n-squared. Algorithms written for a given machine configuration do not need to be rewritten when the machine is expanded.

IPON's programming environment is based on process level data flow which:

- Gives rise to modular programs which can be used as building blocks for more advanced algorithms.

- Reduces any possible communication bottleneck due to the fact that data is only transmitted at the completion of a process as opposed to the completion of an instruction.

- Allows one to exploit inherent parallelism amongst processes.

- The data flow execution paradigm is enforced only upon the processes themselves. Internally, the process can use any other appropriate flow of control paradigm to efficiently execute the algorithm.

- IPON is programmed in a graphical, hierarchical programming language which eases the development problem associated with parallel algorithms.

The optical network, which allows any processor to communicate with any other processor and allows any number of such conversations to take place simultaneously, is diagrammed in (Figure 3). The network consists of n optical transmitters (laser diodes), n acousto-optic deflectors (AOD, Bragg cells) and n photo sensitive receivers (photodiodes). Each processor is attached to one or more transmitters and receivers. The AOD devices serve as beam steerers; they deflect an incoming laser beam at an angle proportional to the frequency applied to the device. For applications where high speed dynamic reconfigurability is not required, low cost mirror based deflection systems based on galvonometers, servomotors, or piezoelectric devices can be used. Connected to this network are a number of homogeneous processors. These processors need not be typical digital processors; indeed one of the motivations behind the development of

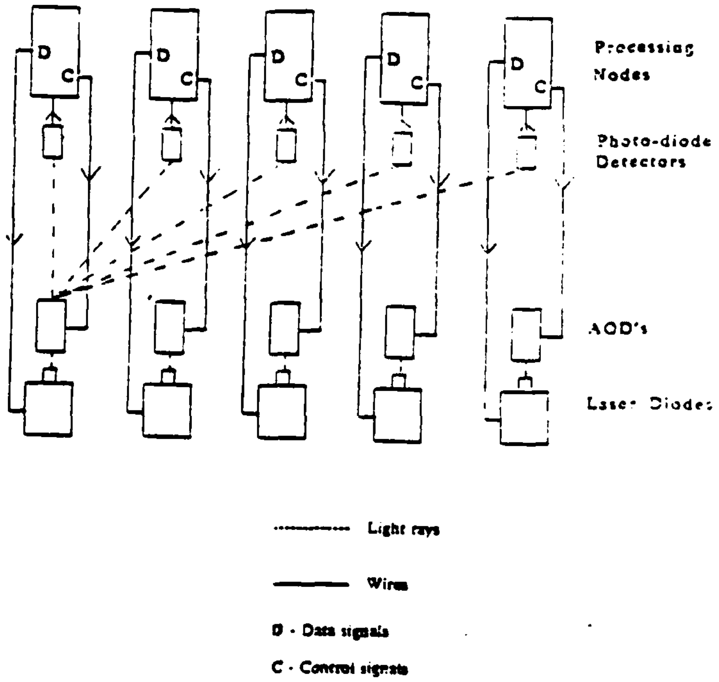


Figure 4-1: Optical Network

IPON was to allow integration of non-traditional image processing devices into a more traditional (in terms of programming and use) image processing system. The reason for this is the fact that digital computers are **not** always the ideal devices for doing image processing. Alternative image processing devices include coherent and non-coherent optical devices [Star82] that enable the computation of complex functions such as the Fourier transform to proceed at the speed of light. Hybrid analog-digital systems [Dood79] have also been developed that perform many image processing functions which, if performed using purely digital techniques, would require orders of magnitude more hardware to produce the same result in the same amount of time. More traditional machines capable of increased throughput, such as SIMD computers, can also be integrated into the IPON system. While many of these approaches are at the present time extremely primitive, the important point is that they can be easily integrated into IPON as the technology matures.

IPON programs will be written in a *graphical* data flow language. The language is also hierarchical, allowing the programmer to view a program at any level of detail he desires. We are choosing to use a graphical language in the hopes that a graphical representation of an algorithm consisting of a number of cooperating parallel processes will be easier to understand, hence easier to construct and debug. It is interesting to note that in most texts describing parallel systems, the system is first represented graphically and then it is shown how to convert this graph to a one dimensional representation, i.e., a program written in a language that supports parallel flow of control operators such as fork and join [Denn76]. While this program retains the same semantics of the original graph, it is no longer as easy to visualize just what function it performs. We feel that it is this *linearizing* of parallel programs, which makes writing and understanding such programs the difficult task that it is today. IPON attempts to reduce this difficulty.

4.3. Current Status of IPON

Substantial progress has been made in the time (approximately one year) since the IPON project was initiated. Some of the accomplishments of the first phases of the IPON effort are listed below:

- Architectural design of IPON.
- Functional emulation of IPON structure.
- Preliminary graphical programming interface.
- Initial investigation of optical network implementation.
- Determination of requirements for distributed control.
- Organization of optical data link interface processor.

Note that most of these areas of research are quite general in nature. Thus, although our immediate objectives relate to IPON, the results obtained with these investigations will be applicable to other multiprocessor and dataflow systems - especially in the areas of optimal resource allocation and scheduling on MIMD and dataflow systems.

We are now in a position to investigate the following aspects of IPON:

- Implementation of prototype optical network.
- Optimal network control and task allocation.
- Use of shared high capacity storage.
- Performance evaluation and optimization.

- Graphical programming system development.
- Hierarchical image database management.
- Integration of special purpose or hybrid processors into IPON.

To date the majority of work has been concentrated in the development of an IPON system emulator. Towards this end a program to construct graphical programs, compile them into an intermediate language and subsequently compile this language has been developed. Furthermore, a primitive version of the task allocator and an interpreter to execute the generated code have also been developed. With these tools, we have developed image processing algorithms in the graphical language and executed them under the IPON emulator on a single processor VAX. However, the emulator currently only supports algorithms which require no iteration or selection and assumes that there exist enough processors to perform all the tasks simultaneously.

Current work is centered around the development and analysis of the optical network. We are constructing a small prototype of the hardware and plan to evaluate the resultant network in terms of speed, reliability, and cost. Furthermore, we are developing the necessary control algorithms through which the processors will interface to the network. After the actual performance parameters of the network are obtained through experiments with the prototype we will run simulations to determine what the actual system throughput would be if a full scale network connecting heterogeneous processors were available.

The simulator will also allow us to investigate various network control and task allocation strategies and determine their effect on overall performance. Once the optimum strategies have been determined we plan to implement them on a network of VAXes and measure the real world performance of such a system. This network of VAXes will initially be connected through the use of a high speed Ethernet, but as development proceeds on the hardware for the optical network the Ethernet will be phased out.

One of IPON's features is the use of heterogeneous processors, each tailored for efficient execution of certain image processing tasks. These processors are interconnected in such a manner that if a portion of a given image processing algorithm can be executed in an extremely efficient manner on a certain processor, then an attempt should be made to execute that task on that processor. Several problems arise when attempting to perform this sort of optimization. One problem is that of measuring what the performance of a processor is when presented with a specific task. The performance of a processor depends on many factors and what is needed is a way of expressing these factors in such a fashion that a task allocator can rapidly determine how well a processor can perform a given task. Another problem concerns the task allocator itself. Even if a processor's performance can be ascertained, the task allocation problem remains a NP-complete problem and heuristics must be used to reduce the time taken to determine task to processor allocation. An algorithm to perform such allocation has been developed but experiments need to be performed to determine its effectiveness.

Development of IPON's programming system is proceeding concurrently with the development of the hardware. The graphical programming language is being expanded to provide a complete set of programming language constructs. The expanded language will allow for the expression of highly parallel image processing algorithms in a manner comprehensible to the programmer. In addition to expanding the language, work is needed in the area of the user interface. This includes determining the most effective manner of interactively manipulating graphical symbols and presenting these symbols in a form which is understandable to the programmer.

To be a usable tool for image processing research, IPON must be able to rapidly access large amounts of data from secondary storage. IPON is designed to operate at real time rates (1 frame every thirtieth of a second). At these speeds, traditional magnetic storage devices form a severe bottleneck. Furthermore, their data storage capacity is somewhat limited. To overcome these problems we are investigating alternative storage technologies such as optical disks. One limitation of optical disks is their write once characteristic. We hope to overcome this problem by using fast temporary bulk store memory as a write buffer. Any temporary changes made to an image will be stored in this memory. Only when the final result has been calculated will the image be written to the optical medium. The software to control such a scheme and simulations demonstrating the resultant increase in performance will be developed.

Hierarchical access to multi-spectral image data at variable resolution, size and resolution is a characteristic of many complex image processing algorithms. IPON will support such access through the use of generic image processing tasks. A generic task will be able to process any size or resolution image. To accomplish this, image access will be provided in terms of an arbitrary number of rectangular sub-images or *segments* which may be configured with respect to one another without altering the actual image data. Images can then be treated as a list of Segment Descriptor Blocks (SDBs) through which image processing tasks access the actual data. Using SDBs, a given image processing task can be written in such a way that it can process a large variety of image formats without need for modification. Research into the question of how to efficiently interpret the SDBs in the IPON environment is to be conducted.

4.4. Conclusions

IPON is meant to be both a tool to design image processing algorithms and a system which can execute these algorithms in real time. We are taking the approach that there exist machines that offer efficient solutions to certain image processing tasks and what is needed is a way to easily and coherently integrate these machines so that they can work together to efficiently execute complex image processing algorithms. Another function of IPON is to demonstrate that digital electronics is not the only way to implement image processing algorithms. The system is to allow experimentation with hybrid digital, analog and optical image processing techniques to determine the advantages and disadvantages associated with such an approach. It is through the use of an *ideal* network, that a system providing the desired capabilities of IPON is possible.

Initial results, both in the design of the software and the design of the network, encourage us to believe that IPON is a viable concept. With further work, IPON will become a flexible, programmer friendly and ultra-high performance image processing system. Such a system has the potential to advance research in fields such as robotics, where the need for an easy to use real time image processing system is large.

Development of the concepts for IPON are nearing completion. Implementation and evaluation remain.

4.5. Image Processing Bibliography

[1] Dennis, J.B. and Horn, E.C. Van,
Programming Semantics for Multiprogrammed Computations.
Communications of the ACM Vol. 9(No. 3):pp. 143-155, October, 1976.

[2] Dood, G. and L. Rossol, ed.,
Computer Vision and Sensor Based Robots.
Plenum Press, New York, 1979.

[3] Etchells, D.
A Study of Parallel Architectures for Image Understanding Algorithms.
Image Understanding Research Report Vol. 104:pp. 133-176, October 19, 1983.
University of Southern California.

[4] Goldwasser, S.
The Image Processing Optical Network: Advanced Architecture for Image PRrocessing.
Computer Science Grasplab report, University of Pennsylvania, July, 1984.

[5] Smitley, D., Goldwasser, S., and Lee, I.
IPON - Advanced Architecture for Image Processing.
In . 12th International Symposium on Computer Architecture, Boston, MA, June, 1985.

[6] Stark, H., ed.,
Applications of Optical Fourier Transforms.
Academic Press, New York, 1982.

9. List of Publications Directly Related to This Project:

Surface Description From
Vision and Touch
Peter Allen
Precented at the 1st
Int. IEEE Conf. on Robotics
Atlantic, March, 84'.

Object Recognition Using
Vision and Touch
Peter Allen
Ruzena Bajcsy
Submitted to IJCAI, 85'.

Sensing Strategies
Ruzena Bajcsy
Peter Allen
Presented in the 2nd ISRR,
Tokyo, Japan, August, 84'.

Stereo Processing of
Aerial Images
Ruzena Bajcsy
David L. Smitley
Proceedings of the 7th
Int. Pattern Design Conf.,
Montrial, August, 84'.

Final Report: The Design and
Construction of a Four Degree
of Freedom Camera Controller
John F. Summers
Internal Paper

Sphere Packing Algorithm
For Sparse 3-D Points
Ruzena Bajcsy
Franc Solina
Internal Paper

Computational Models of
Visual Hyperacuity
Eric Paul Krotkov
Submitted to the
Journal of CUBIP

Grasp Lab Memo:
Construction of a Three
Dimensional Surface Model
Eric Krotkov
Internal Paper

An Architecture for the Real-Time
Display and Manipulation of
Three-Dimensional Objects
Samuel Goldwasser
R.A. Reynolds
Internal Paper

A Generalized Object Display
Processor Architecture
Samuel Goldwasser
Internal Paper

GRASP:NEWS
Quarterly Progress Report
The GRASPlab:
General Robotics and
Active Sensory Processing
Group
School of Engineering and
Applied Science
University of Pennsylvania
Philadelphia, PA 19104

10. GRASP Lab Memos:

What Can We Learn From
One Finger Experiments?
Ruzena Bajcsy, David Brown
Jeff Wolfield, Dan Peters
Technical Report MS-CIS-83-03
Grasp Lab 01

This paper describes results from recent experiments performed in our laboratory with two different tactile sensors attached to a cartesian coordinate system. The one sensor is in the form of a flat surface equipped with an array of 8 by 8 strain-gage sensors on loan from the Lord Corp., USA. The other sensor is in the form of a rigid finger of an octahedron tapered with four sides and ended with one tip. All together the device has 133 pressure sensors. This device was obtained as part of US-French collaboration from LAAS Toulouse, France (Dir., Prof. G. Giralt).

We shall report results on calibration , on physical properties of the sensors, limitations on spatial resolution and pressure sensitivity. We have investigated the classificatory power with respect to material hardness, elasticity and the surface texture. Finally, we outline the open problems and the near future plans.

Tactile Information Processing
The Bottom Up Approach
Working Paper
Ruzena Bajcsy, Greg Hager
Technical Report MS-CIS-83-38
Grasp Lab 09

A primal sketch for tactile information processing is outlined. It is further argued that from the basic three primitives: hardness, surface normals, and local curvature all other tactile features can be constructed.

The Recognition and Representation of 3D Images for
A Natural Language Driven Scene Analyzer
Amy Elizabeth Zwarico
MS-CIS-84-29
GRASP LAB 20

Two necessary components of any image understanding system are an object recognizer and a symbolic scene representation. The LandScan system being designed is a query driven scene analyzer in which the user's natural language queries will focus the analysis to pertinent regions of the scene. This is different than many image understanding systems which present a symbolic description of the entire scene regardless of what portions of that picture are actually of interest. In order to facilitate such a focusing strategy, the high level analysis which includes reasoning and recognition must proceed using a top-down flow of control, and the representation must reflect the current sector of interest. This thesis proposes the design for a goal-oriented object recognizer and a dynamic scene representation for LandScan - a system to analyze aerial photographs of urban scenes. The recognizer is an ATN in which the grammar described sequences of primitives which define objects and the interpreter generates these sets of primitives. The scene model is dynamically built as objects are recognized. The scene model represents both the objects in the image and primitive spatial relations between the objects)

A New Approach to Robotic Approach
Tactile Perception
Ruzena Bajcsy
Kenneth Y. Goldberg
MS-CIS-84-31
GRASP LAB 02

Psychologists believe that tactile perception involves receptors located in both the skin (cutaneous) and the joints (kinethetic). Research in the area of robotic tactile perception has focused on cutaneous sensors, producing tactile grids with increasingly improved resolution. A robot developed at the University of Pennsylvania, however suggests that the most efficient way to achieve tactile recognition is to process kinesthetic information. This approach has implications for both psychology and industry.

On Grasping With A Three-Fingered
Mechanical Hand
J. M. McCarthy
MS-CIS-84-32
GRASP LAB 03

This paper uses the generalized theory of screws to formulate the problem of grasping a general solid object between the three finger tips of a mechanical hand, and securely holding the object against the action of an arbitrary set of externally applied forces. A condition on the geometry of the grasp is presented which assures that the allowed motion of the object can be opposed by frictional contact forces, and it is shown that any such grasp can be broken by applying the proper choice of external forces. The magnitude of this additional loading is a measure of the quality of the grasp.

Computer Architecture for Grasping
Samuel M. Goldwasser
MS-CIS-84-33
GRASP LAB 04

The Integrated Tactile Network Architecture or ITNA is a hierarchical system for managing the interaction of tactile sensing and motor control in the 3-D active sensory environment. The overall ITNA includes custom dedicated hybrid front end tactile arrays incorporating electronics and microprocessors for sensor linearization, tactile information preprocessing, and local feature extraction; approaches to the distributed motor control of manipulator fingers for grasping; interconnection networks for guarded movement and reflex arcs; and special purpose hardware for model generation derived from tactile information. This paper primarily addresses the overall ITNA structure and, in particular, the design of an intelligent sensor array and its associated communications subsystem.

Feeling By Grasping
Ruzena Bajcsy
Michael McCarthy
Jeffrey C. Trinkle
MS-CIS-84-34
GRASP LAB 05

This paper specifies constraints based on the geometry of the grasped object, on geometry of the hand and the kinematics of the constrained object which determines how to grasp an object.

Surface Descriptions From Vision
and Touch
Peter Allen
MS-CIS-84-35
GRASP LAB 06

The goal of vision is object recognition. Recent research has shown that the human visual system creates a surface description of a scene, including depth and orientation information at all points in a scene as a first step before creating an object centered 3D description. This description is referred to as a 2 1/2 D sketch.

Machine vision systems presently do not have the capability of creating this 2 1/2 D sketch from visual information alone, especially for curved surface objects. By using tactile data in cooperation with vision, a method is proposed for creating a surface description of an object. This surface description uses bicubic surface patches as a primitive.

Once a surface sketch is created with bicubic surface patches the next steps in the hierarchy of processing are feasible, including a transformation to a full 3D object centered description.

Stereo Processing of
Aerial Images
Ruzena Bajcsy
David L. Smitley
MS-CIS-84-36
GRASP LAB 07

An Architecture for the Real-Time Display and Manipulation of Three-Dimensional Objects
S.M. Goldwasser
R.A. Reynolds
MS-CIS-84-37
GRASP LAB 08

A special purpose multiprocessor architecture has been developed which facilitates the high speed display and manipulation of shaded three dimensional objects or object surfaces on a conventional raster scan CRT. The reconstruction algorithms exploit the capability to divide object space into totally disjoint cubical regions permitting multiple display processors to access independent memory banks concurrently without describing rotation, translation, and scaling are incorporated into one short table facilitating very rapid manipulation of the image display presentation.

A Generalized Object Display Processor Architecture
S.M. Goldwasser
MS-CIS-84-38
GRASP LAB 10

A multiprocessor architecture has been developed which addresses the problem of the display and manipulation of multiple shaded three dimensional objects derived from emperical data on a raster scan CRT. Fully general control of such parameters as position, size, orientation, rotation, tone scale, and shading are accomplished at video rates permitting real-time interaction with the display presentation.

The GODPA architecture is based on a large number of relatively simple processing elements which access their own memory modules without input conflict. Reconstruction algorithms are used which do not require any complex arithmetic or logical high speed operations. This hardware organization is highly modular and expandible and is ideally suited for implementation with VLSI technology.

Page Composition of Continuous Tone Imagery
Samuel M. Goldwasser
Donald E. Troxel
MS-CIS-84-39
GRASP LAB 11

A system has been developed which represents an effective unified framework for interactive layout and page generation of pictures and linework for applications in the graphic arts. The functional structure and logical organization of this system are based upon the segment display processor (SDP) architecture which offers a generalized approach to the manipulation of multisegment multiformat data. Interactive layout is accomplished with the aid of a graphics digitizing tablet and proof TV display. Subsequent input scanning, sizing, enhancement, and merge operations are fully automatic. The system handles arbitrary shaped regions, type-on-tone, and optimally codes areas of continuous tone and line art copy individually. The PAGES system described in this paper is centered around a software emulation of the SDP supporting continuous tone imagery and scanned type. Most of the effort devoted to this development will be directly applicable to an SDP-based system implemented with the aid of special purpose high-speed hardware in the future.

Grasp Lab Memo:
Construction of a Three
Dimensional Surface Model
Eric Krotkov
MS-CIS-84-40
GRASP LAB 12

This report describes the structure and construction of an initial intermediate level surface modelM_0 subject to the criteria and constraints imposed by the domain of stereo aerial imaging. M_0 is a surface-based polyhedral network supplemented with relations. It is constructed from data derived from low level vision processes, and thus can be aptly called "bottom-up" or "data-driven." At the same time the representation is useful for high-level "top-down" processing. The model is built on top of Surf*UP* and has been fully implemented and documented, including: routines to compute relations on faces; and a high level driver program.

Quarterly Progress Report
Volume 2, No. 1
MS-CIS-84-41
GRASP LAB 13

Active Touch and Robot
Perception
Ruzena Bajcsy
Kenneth Y. Goldberg
MS-CIS-84-42
GRASP LAB 14

Psychologists distinguish between active and passive touch. The latter arises when objects are brought into contact with a passive tactile surface, such as the palm of the hand. Active touch describes a dynamic exploration of objects involving receptors located in both the skin (cutaneous) and the joints (kinesthetic). Research in the area of robotic tactile perception has focused on passive touch, developing cutaneous grids with increasingly improved resolution. A robot developed at the University of Pennsylvania, however,suggests that the most efficient way to achieve tactile recognition is to process kinesthetic information gained from active exploration. The results may be of interest to researchers in both psychology and robotics.

Computational Models of
Visual Hypercuity
Eric Paul Krotkov
MS-CIS-84-43
GRASP LAB 15

The process of visual hypercuity is described and analyzed in the terms of information theory. It is shown that in principle, the detection and representation of both luminance and edge features can be performed with a precision commensurate with human abilities.

Algorithms are formulated in accord with the different representational method and are implemented as distinct computer models, which are tested with vernier acuity tasks. The results indicate that edge information encoded either in the manner proposed by Marr and his colleagues (as zero-crossings in the Laplacian of a Gaussian convolved with the image) or when encoded as a simple filtered difference allows finer spatial localization than does the centroid of the intensity distribution.

In particular it is shown that to judge changes of relative positions with a precision of 0.1 sec arc in two and three dimensions, it is sufficient to represent the displacement of an edge by the difference of two Laplacian-Gaussian filters rather than by the difference between interpolated zero-crossings in them. This method entails no loss of relative position information (sign), allows recovery of the magnitude of the change, and provides significant economies of computation.

Sphere Packing Algorithm
For Sparse 3D Points
Ruzena Bajcsy
Franc Solina
MS-CIS-84-44
GRASP LAB 16

An efficient way for representing objects defined by sparse 3-D surface points is described. Data collected by stereo and range imaging techniques can be considered as an approximation of surfaces used for volume representation by packed nonoverlapping spheres. The technique described here is a modification of the algorithm introduced by R. Mohr and R. Bajcsy in Packing Volume by Spheres, IEEE Transactions on PAMI-5, pp. 111-116, 1983.

PAMH Coordinate System
Jeffrey C. Trinkle
MS-CIS-84-45
GRASP LAB 17

This paper is meant to document the various coordinate systems used in PAMH. The matrix transformations are all defined, and their use is described.

PAMH Micro Guide
Ed Walsh
MS-CIS-84-46
GRASP LAB 18

Quarterly Progress Report
Volume 2, No. 2
MS-CIS-84-47
GRASP LAB 19

Angy: A Rule-Based Expert
System For Identifying And
Isolating Coronary Vessels
In Digital Angiograms
S. A. Stansfield
MS-CIS-84-49
GRASP LAB 21

This paper presents work being done in the development of a rule-based expert system for identifying and isolating coronary vessels in digital angiograms. The system is written in OPS5 and LISP and uses low level processors written in C. The system embodies both stages of the vision hierarchy: The low level image processing stage works concurrently with edges (or lines) and regions to segment the input image. Its knowledge is that of segmentation, grouping, and shape analysis. The high level stage then uses its knowledge of cardiac anatomy and physiology to interpret the result and to eliminate those structures not desired in the output.

The Image Processing Optical
Network: Advanced Architecture
For Image Processing
Samuel M. Goldwasser
MS-CIS-84-50
GRASP LAB 22

The Image Processing Optical Network (IPON) is an ultra high performance architectural framework being developed to support image acquisition, low and medium level image processing and analysis, image display, and image storage using digital and hybrid technology. IPON assumes the use of the technology of the 1990s and beyond including hybrid optical systems and other novel devices which depart from the strictly 'more gates on semiconductor' philosophy of the past 20 years.

IPON will be an MIMD network utlilizing non-homogeneous functional nodes of a variety of types. It will be dynamically partitionable and reconfigurable using a non-blocking optical interconnection network.

IPON will support the use of optical-hybrid technology for key components to provide high bandwidth communications, high capacity buffering, and certain types of high speed processing. User level programming of IPON will be accomplished using the concept of process level dataflow control via an interactive Graphical Image Processing Language.

This paper outlines some initial thoughts on the organization and implementation of IPON.

A Programming System For Distributed Real-Time Applications
Insup Lee
MS-CIS-84-51
GRASP LAB 23

A distributed programming system designed to support the construction and execution of a real-time distributed program is presented. The system is to facilitate the construction of a distributed program from sequential programs written in different programming languages and to simplify the loading and execution of the distributed configuration language. The language is used to write the configuration of a distributed program, which includes resource requirements, process declarations, port connections, real-time constraints, process assignment constraints, and process control statements.

Final Report: The Design and Construction of a Four Degree of Freedom Camera Controller
John F. Summers
MS-CIS-84-52
GRASP LAB 24

A system has been developed that controls the positioning of a pair of camera to provide the possibility of active visual sensing. The system uses an 8085 based microprocessor to monitor and update the position of the platform upon which the cameras are mounted. The camera platform possesses four degrees of freedom: up and down, side to side, pan motion, and tilt motion. The microprocessor is interfaced to a VAX 11/750 such that programs running on the 750 can control the platform motion. A series of higher level manipulation programs have been written and installed as user utilities on the VAX, running under Unix. These utilities the system user to place control of camera positioning into the hands of image processing software. This closes the camera positioning feedback loop, and should lead to the development of autonomous "intelligent" camera manipulation routines.

PAMH Control Systems
Jeffrey C. Trinkle
MS-CIS-84-53
GRASP LAB 25

This report describes the position control algorithm, in joint coordinates, and the grip pressure controller used by the PAMH system. The linear analysis of the position controller is given here. The results are currently being used to determine the gains of the position controller. The grip pressure controller presented is not yet implemented.

Converging Disparate
Sensory Data
Ruzena Bajcsy
Peter Allen
MS-CIS-84-54
GRASP LAB 26

Object recognition systems using single sensors (typically vision) are still limited in their ability to correctly recognize different three dimensional objects. By utilizing multiple sensors (in particular, vision and touch) more information is available to the system. This paper is an attempt to show the utility of multiple sensors and explore the problems and possible solutions to converging disparate sensory data for object recognition.

A New Development in Camera
Calibration -Calibrating a Pair
of Mobile Cameras
Alberto Izaguirre, Pearl Pu, and
John Summers
MS-CIS-84-55
GRASP LAB 27

Pennsylvania Articulated
Mechnical Hand An End
Effector To Determine
Shape By Touch
MS-CIS-84-56
GRASP LAB 28

This paper provides a description of the Pennsylvania Articulated Mechanical Hand (PAMH)*, a mechanical hand with independent joint control to be used for object recognition. PAMH is currently being developed at the University of Pennsylvania. The mechanical design of PAMH is presented and the piezo plastic tactile sensor used to determine spatial resolution is detailed.

Integrating Vision and Touch For Grasping of an Object*

Ruzena Bajcsy
MS-CIS-84-57
GRASP LAB 29

The aim of this paper is to present considerations that go into the design of a system tha

For the visual sensor we assume that we have available stereo cameras or their equivalent. As the tactile sensor we use an articulated multifingered hand equipped with tactile sensory arrays as the data acquisition device. We shall present available configurations of these devices. Then we shall investigate the sensory processing, in particular what representation schemas should be considered. We shall argue that the observer-centered representation as opposed to the object-centered representation is more important for grasping. Finally, a rule based schema for the control strategies will be outlined. As examples, first some artificial geometric objects than some real laboratory objects from the blocks world will be analyzed.

Angy: A Rule-Based Expert System For Identifying and Isolating Coronary Vessels in Digital Angiograms

Master's Thesis
MS-CIS-84-63
GRASP LAB 30

This thesis details the design and implementation of ANGY, a rule-based Expert System in the domain of medical image processing. Given a subtracted digital angiogram of the chest, ANGY identifies and isolates the coronary vessels, while ignoring any non-vessel structures which may have arisen from noise, variations in background contrast, imperfect subtraction, and non-relevant anatomical detail. The over all system is modularized into three stages: The preprocessing stage and the two stages embodied in the expert itself. In the preprocessing stage, low level image processing routines written in C are used to create a segmented representation of the input image. These routines are applied sequentially. The expert system is rule-based and is written in OPS5 and LISP. It is separated into two independent stages: The low level image processing stage embodies a domain independent knowledge of segmentation, grouping, and shape analysis. Working with both edges and regions, it determines such relations as parallel and adjacent and attempts to refine the segmentation begun by the preprocessing. The high level medical stage embodies a domain dependent knowledge of coronary physiology and anatomy. Applying this knowledge to the objects and relations determined in the preceeding two stages, it identifies those objects which are vessels and eliminates all others.

Filling in the Gaps: A Computational Theory of Contour Generation
David J. Heeger
MS-CIS-84-64
GRASP LAB 31

The problem of contour generation is posed as an example of perceptual organization. A computational framework is presented which is a uniform theory of contour generation. The same computational process derives contours of all different types of subjective boundaries. The basic process of contour generation is to fill in the gaps in contours. Psychophysical experiments on subjective contours are used to constrain the process of filling in the gaps. The algorithm demonstrates the feasibility of computing global properties (contours) from purely local computations.

Detecting Tactile Feature Points with a Robot Hand
Kenneth Y. Goldberg
Edward S. Walsh
MS-CIS-84-65
GRASP LAB 32

Changes in edge curvature can be detected by applying differential operators to a list of boundary points. Such changes, or *feature points*, provide a representation for object shape which is well-known in machine vision. We apply this technique to sparse tactile data, using the Pennsylvania Articulated Mechanical Hand to discriminate between six sample objects.

12. APPENDICES

I. Stereo Processing of Aerial Images

II. LandScan: A Natural Language and
Analyzing Aerial Images

III. LandScan: A Computer Vision Syste

IV. Recognition and Representation of 3D Objects for LandScan - A
Natural Language Driven Scene Analyzer

V. Implementation of a Gaussian-Smoothing Gradient-Based Edge
Detector

Recognition and Representation of 3D Objects for LandScan - A Natural Language Driven Scene Analyzer

Amy Zwarico

CIS Dept/D2
University of Pennsylvania
Philadelphia, Pa 19104

Image Representation, Vision Systems, Object Recognition

short paper

[Radack, et al 84]
Radack, Korein, Ganis, McNally, Korein, Shapiro. *NASA Programmer's Guide* CIS Department, University of Pennsylvania, 1984.

[Shapiro 84] Shapiro and Haralick. A Heirarchical Relational Model for Automated Inspection Tasks. In *Int. Conf. on Robotics, Atlanta, Ga.*. 1984.

[Sloan 81] Kenneth R. Sloan , P. G. Selfridge. Reasoning about images: Applications to aerial image understanding. In *Proc. 1981 Image Understanding Workshop*, pages 1-6. April 1981.

[Smitley 84] David L. Smitley , Ruzena Bajcsy. Stereo Processing Aerial, Urban Images. In *Seventh International Conference on Pattern Recognition*, pages 432-436. July 30-August 2, 1984.

[Talton 84] David Talton. *Implementation of a Gaussian-Smoothing Gradient-Based Edge Detector*. Technical Report MS-CIS-84, University of Pennsylvania, 1984.

[Zwarico 84] Amy Zwarico. *The Recognition and Representation of 3D Images for A Natural Language Driven Scene Analyzer*. Technical Report MS-CIS-84-29, University of Pennsylvania, 1984.

Abstract

Two necessary components of any image understanding system are an object recognizer and a symbolic scene representation. The LandScan system currently being designed is a query driven scene analyzer in which the user's natural language queries will focus the analysis to pertinent regions of the scene. This is different than many image understanding systems which present a symbolic description of the entire scene regardless of what portions of that picture are actually of interest. In order to facilitate such a focussing strategy, the high level analysis which includes reasoning and recognition must proceed using a top-down flow of control, and the representation must reflect the current sector of interest. This paper proposes the design for a goal-oriented object recognizer and a dynamic scene representation for LandScan ; a system to analyze aerial photographs of urban scenes. The recognizer is an ATN in which the grammar describes sequences of primitives which define objects. The Scene Model is dynamically built as the objects specified by the queries are recognized. Thus the control of the scene modelling is top-down, reflecting the user's interest in the scene. The Scene Model represents both the objects in the image and primitive spatial relations between these objects.

1. Introduction

LandScan (LANguage Driven SCene ANalyzer) is a goal-oriented computer vision system which uses natural language to drive the scene analysis of 3D images of aerial views of urban scene. Goal oriented scene analysis restricts the analysis to those areas of the scene which are currently of interest to the user of the analyzer. Both recognition and modelling are driven by user queries. Answering these queries will require the following reasoning operations to be facilitated:

1. determining the existence of an object
2. finding an object part
3. determining locative relations, both simple and complex, among objects.

The object recognition paradigm allows the first two operations to be performed. The Scene Model - objects and the relations between objects - allows previously recognized objects to be referenced and determines the locative relations among objects.

This paper will propose a solution to the problem of goal driven recognition and representation of 3D objects in aerial views of urban scenes to be used by a language driven scene analyzer. An Augmented Transition Network (ATN) has been chosen to perform the object recognition because it has a top-down flow of control which facilitates the interface between the queries and the recognition process. It also represents in a perspicuous manner the global knowledge necessary for recognizing objects in this domain. A dynamic scene model will be generated incrementally as the user focusses the image analysis to areas of the image which are of interest to him/her. The Scene Model symbolically represents the objects which have been recognized and the primitive spatial relations which hold between them.

First, the related work will be surveyed. Then a brief overview of the LandScan system will be presented. Next the design of the ATN used for object recognition will be discussed. Following this the Scene Model will be described.

2. Related Work

A large corpus of research on aerial image understanding *per se* exists, [Harlow 84], [Hwang 83], [Nagao 79], [Sloan 81], [Quam 78], [Faugeras 81], [Glicksman 83], [Reynolds, et al 84], [Potmesil 83]and many general vision techniques are applicable to the aerial domain. Large aerial projects have been undertaken at USC [Nevatia 83]and at SRI [Fischler 83]

The 3D MOSAIC scene understanding system [Herman 83]incrementally derives a scene model of an aerial view of an urban scene. Like LandScan,the scene model is dynamic - constructed incrementally as more data becomes available. Domain-specific knowledge is used to help identify incomplete objects. A fundamental difference is that the construction of the Scene Model in LandScan is goal driven and reflects the user's interest in the scene.

ATN's have been used almost exclusively in natural language processing [Bates 81], [Winston 79], [Winograd 83]. A notable exception to the use of ATN grammars for natural language understanding is the system designed by Tropf and Walter [Tropf 83] which uses an ATN model for the recognition of 3D objects with known geometries. The recognition process performed by their system is one of "analysis-by-synthesis" in which hypothetical model instances (prototypes) are generated and then verified by the ATN. The verification process compares the prototypes to the actual 3D data using the ATN. If the similarity between the prototype and the image exceed some threshold then the prototype is considered to be a model instantiation of the actual data.

Shapiro and Haralick [Shapiro 84] describe a hierarchical, relational 3D model which is influential in our design. Their model provides precise, accurate information to be used by low-level vision and inspection processes as well as information required by high-level vision and reasoning processes. All of the information is represented by using "spatial data structures", each consisting of a recursive set of relations. The hierarchy consists of four levels: world, object, part, and surface/arc.

Rosenthal [Rosenthal 81] proposed a model and interpreter for analyzing aerial images of urban settings. In some ways, Rosenthal's work was the impetus for this system. He proposed a purely hierarchical model of the world which is ordered by the ON relations and a goal driven production system for the recognizer. It has a database which contains descriptions of objects and regions. He introduced an Object Description Notation to encode a set of descriptors which would be adequate for the system to describe an object in the scene.

The work of Talmy and Herskovits [Talmy 83] [Herskovits 82] [Herskovits 84] has influenced the design of both the topological relations in the models and the choice of linguistic attributes which must be associated with objects in order to insure a robust and reliable natural language interface. It is from their work that the need for a single meaning for a single relation was discovered.

3. LandScan

LandScan is a query driven scene analyzer for 3D aerial views of urban settings which uses low, middle and high level vision processes, high level reasoning and natural language understanding to analyze an image. The low level image processing routines detect edge points (edgels), perform stereo matching to obtain the 3D information crucial to the higher level analysis, and segment the picture into various picture primitives - edges and regions. The middle level imaging modules add topological properties to the regions detected by the low level routines creating a Surface Model. The high level vision uses the regions and topological properties of the Surface Model along with *a priori* knowledge of the domain to identify a subset of these regions as an object. Operating simultaneously with the recognizer is a modeller which creates a model of the scene which facilitates high-level reasoning. Finally, the natural language interface and high level reasoner parse queries, search the image for the data in question, and using the world and object models generate the appropriate response to queries.

All low and middle level image processing is performed in a bottom-up fashion when the digitized image is presented to the system for analysis. The high level vision uses an ATN driven by the queries parsed by the natural language interface to recognize objects and build the Scene Model. No high level recognition or analysis is performed until a query is made. When a question is asked, the Scene and Surface Models are analyzed only as much as is necessary to enable the generation of an adequate response for the user. Only those objects expressly mentioned in queries are represented in the Scene Model. Although using this recognition strategy will increase the time necessary to answer a question, it will probably reduce the overall amount of work which is done in analyzing a scene. The system will not analyze the entire scene, only those areas of interest to the user.

Each object in the scene is represented by a labelled set of polyhedra. The polyhedra was chosen as the primitive for the representation of objects. This choice was not arbitrary, but carefully considered in the system domain, aerial photographs of urban scenes. Looking at such images they appear to be composed of polyhedra of various sizes and shapes at different distances from the ground. Both the Scene and Surface Models are implemented in SurfsUP [Radack, et al 84].

In order to perform the high level vision and reasoning tasks required by this system, world knowledge must also be encoded into the system. This global knowledge will be used to recognize objects, understand natural language queries, and search the Scene Model to obtain an answer to a query. Presently, three sources of *a priori* knowledge have been determined necessary to perform the above mentioned tasks. They are represented by an ATN grammar which describes the manner in which surfaces are grouped to form objects, a World Model and an Object Model. The World and Object Models are very similar to those in the Shapiro and Haralick [Shapiro 84] system as well as Rosenthal's Conceptual Hierarchy [Rosenthal 81]. Like the hierarchical relational model of Shapiro and Haralick, the World Model describes the features and relations of the objects in the domain. The objects are those which can be expected in an urban scene - buildings, streets, sidewalks, etc. The Object Model represents the expected physical features and linguistic properties of the objects in the domain (object parts and those features which affect the usage and interpretation of a spatial constructs - phrases describing the spatial relations between objects) [Talmy 83], [Herskovits 82], [Herskovits 84].

4. Object Recognition

An Augmented Transition Network (ATN) has been chosen as the paradigm for object recognition in LandScan. It is composed of three parts: the grammar, a dictionary, and an interpreter. The grammar represents the *a priori* or world knowledge that the system must have in order to recognize objects and assign them "cultural" labels (ie. building, street). The dictionary, Surface Model, represents the actual data which will be used in the recognition process. The third component of the recognizer is the Lisp program which provide the control structure for the process. The syntactic approach has been adopted despite earlier failures [Fu 82]. Earlier attempts using a syntactic approach failed because the primitives which were combined were too low level (edges, etc) and the matching sequences were too strict. In LandScan, the primitives used are higher level (surfaces) and thus have more information associated with them. The grammar rules in LandScan do not specify a strict matching sequence. Instead they specify the properties which must hold between the simpler components of an object. Since the rules are more general - there are fewer in the system thus simplifying the recognition process. The following sections will justify the use of an ATN to perform object recognition and discuss the three components comprising the recognizer - the world knowledge which is represented by an ATN grammar, the visual data or dictionary, and the ATN interpreter which drives the recognition strategy.

4.1. Justification for the Use of an ATN

The goal of the LandScan system is to perform query driven analysis of urban scenes. This places two constraints on the object recognition process: it must have a top-down control structure, finding only those objects references in the query, and it must encode global knowledge about a domain in which objects of the same type may have diverse appearances.

The entities found in an urban scene fall into several general categories - buildings, streets, sidewalks, vehicles, and fields to mention a few. Although the objects in the domain are known, their appearances cannot be precisely predicted. The ATN formalism enables the global knowledge about object appearances to be encoded as a generative model (grammar) for constructing objects from the primitives in the scene while driving the recognition in a top-down fashion.

Although the ATN provides both the top-down control structure as well as the representation of the global knowledge necessary to perform object recognition in this domain, the inherently linear ordering it places on the scanning of input does not seem appropriate for a vision process. We do not obtain the data from a scene one "element" at a time, nor is it likely that we match the features which we have learned to associate with an object in a specific order. Instead, it is likely that we match on "prominent" features in the visual data. Despite these fundamental differences, the ATN is appropriate to use in recognition. The problem with using "prominent features" is that it is difficult if not impossible to model these features into a system. Thus we must fall back to describing an object in terms of the primitives which define it. The ATN grammar presents a straight forward description of a sequence which generates an object from a set of visual primitives.

Thus an ATN is appropriate because it has a top-down control structure, a straight-forward description of the global knowledge necessary for performing object recognition, and separates the control structure from the grammar simplifying modification of the recognizer.

4.2. The Object Recognition Grammar

The ATN grammar represents the world knowledge necessary to enable object recognition. It is a generative model describing in a sequential manner the set of faces and the relations between these faces which must appear in the Surface Model in order to recognize a particular object.

The grammar as written is a set of two level networks. This is considerably simpler than most ATN's which handle natural language utterances. Each network is represented by a set of grammar rules. The bottom level concerns itself with the recognition of "simple objects." An object is simple if its further decomposition into parts will result in no entity which is in the domain of objects. For example, decomposing a building with a pitched roof will result in two halves of a pitched roof. Neither of these entities are considered objects in the domain - they are parts of objects. This level consists of the networks SIMPBUILD, SIMPSTREET, SIMPFIELD, and SIMPSIDEWALK. The top level combines the simple objects which were recognized in the first level of the network into "complex objects". A complex object is decomposable in a nontrivial way into at least one simple object. The top level networks are BUILDING, STREET, FIELD, and SIDEWALK.

A network is a set of nodes and arcs. The nodes represent how far the system has progressed in the object recognition (the state of the computation). The arcs represent the patterns (object primitives of simple objects) which must be matched in order to proceed further in the recognition of that particular object.

The states have two part names [Bates 81]. The first part of the name indicates the name of the network and the second part describes either how far along this state is in the recognition process or the subtype of the object being recognized.

The arcs are represented by lists of the form (TYPE HEAD TEST ACTION). TYPE indicates the category or arc type. The possible arc types in this system are:

- PUSH - call to a "simple" network
- CAT - search the dictionary (Surface Model) for an appropriate face
- POP - return to a calling network or add the recognized object to the Scene Model
- JUMP - go to the next state without searching for a primitive object or face

HEAD is dependent upon the arc type. HEAD can be a syntactic category - words or lists of words, a constituent type, the next state, or the form in which the data "parsed" is to be returned. TEST is a list (possibly empty) of tests to be performed before the arc can be traversed. The tests specify the relations which must hold between various comonents of the object, provide further checking of the features of a component, and provide context sensitivity. ACTION is a list of actions to be performed as the arc is traversed. The possible register setting and structure building actions are:

- (SETR REG VALUE) - sets the register REG to the evaluation of VALUE
- (SETRQ REG STRING) - sets the register REG to the literal STRING
- (ADDR REG VALUE) - appends the evaluation of VALUE to the end of the list in REG
- (BUILDQ <OBJECT_TYPE OBJECT SUBTYPE>) - builds an object instance
  - OBJECT_TYPE is a major object type
  - OBJECT is the OBJECT register
  - SUBTYPE is the SUBTYPE register

There are two registers associated with the system - a SUBTYPE register and an OBJECT register. The SUBTYPE register s a feature register whose value is a string indicating the subtype name of an object. The OBJECT register is a role register containing a list of all the faces which comprise the object. As faces are found which match the generative sequence described by the grammar, they are added to the OBJECT register.

4.3. The Dictionary

The dictionary in the recognizer is the Surface Model which represents both the geometric and topological information about the surface primitives in the scene. [Krotkov 84]

The geometric attributes associated with each surface or face are the area of the face, its surface normal, centroid, shape, type and compactness. The relations between the faces represent the topological properties of the faces. In order to do recognition six topological relations are necessary - ADJACENT, CONTIGUOUS, LOOKSADJACENT, LOOKSCONTIGUOUS, ABOVE, and CONTAINS. Two faces are considered ADJACENT if they share at least one point. Two faces are CONTIGUOUS if they share at least two points - a segment. The LOOKSADJACENT and LOOKSCONTIGUOUS relations hold if the ADJACENT and CONTIGUOUS relations respectively hold between the two faces projected onto the x-y plane. A face is considered ABOVE another face if the z coordinate of the centroid of the first face is strictly greater than the z coordinate of the centroid of the second face. The CONTAINS relation means that one face is completely surrounded by another face when they are both projected onto the xy-plane.

These topological relations are represented by adjacency matrices - one matrix per relation. The "nodes" in the graph represented by the adjacency matrix are the faces which have been found in the scene. The matrix is an n x n boolean array where 0 corresponds to no relation between faces and a 1 to the relation holding between them (n is the number of faces in the surface model). None of the relations are reflexive. ABOVE and CONTAINS are transitive, ADJACENT, CONTIGUOUS, LOOKSADJACENT and LOOKSCONTIGUOUS are symmetric.

4.4. The Control Structure

Unlike most ATN's (in natural language understanding as well as other applications) which have been designed to parse an input string, this system will not have an object which it wishes to parse into its components in order to confirm that it is a valid object. This ATN interpreter operates as a generator taking a grammar and a dictionary as input and producing strings as output. The output string from the object recognizer is an object instance.

The control structure for the ATN is provided by a generator which is a series of LISP functions. The first function GENERATE is called with one argument - the starting state of the grammar. The initial configuration (initial state, register list, stack) is set up by this call. A function ATN is then called with the starting state. ATN is the function which selects the arc which is to be followed. The backtracking is a simple, depth first strategy. If the first arc fails then the next arc in the arc list associated with the state is called. From ATN the EVALARC function is called with the arc to be evaluated and the association list representing the set of registers. First EVALARC determines the type of arc which is being considered as a possible transition. Once the arc type has been determined, the function and the tests associated with that arc are performed. Finally, if all the tests are true, the actions are evaluated by the EVALACTIONS function and the ATN enters a new state and the traversal continues in this fashion until a final state is reached or the functions gets halted in a non-final state.

5. Representation - The Scene Model

The final representation for the scene must facilitate the operations necessary for high level scene analysis to be performed. These actions include determining the relations, both complex and simple, among objects; and locating and identifying specific objects and object parts. Since the analysis is query driven a dynamic Scene Model was designed which facilitates these operations, paying special attention to the representation of the primitive spatial relations between objects. A dynamic Scene Model is one to which information can be added to it as further image analysis occurs. The scene model in 3D MOSAIC [Herman 83] is also incrementally derived as more data becomes available but the modelling process is data driven. LandScan builds a model using a query driven control. In other words, the modeller obtains more data as the user directs the vision system to analyze other areas of the scene which are of interest to him/her. Thus the Scene Model reflects the user's interest in the scene. The LandScan dynamic scene model is especially useful because it is flexible. The accuracy of the scene model increases as new data is acquired. Thus old hypotheses can be discovered false, deleted, and the scene model updated to reflect the more accurate understanding of the scene. In LandScan, when the scene analysis of a new image begins the scene model is empty. As questions are asked, the scene analyzer/constructor searches for the entities whose existence is in question using the object recognizer described above. As soon as the objects queried are found they are added to the the Scene Model. Thus the Scene Model also reflects the history of the user's interest in the image. The Scene Model is composed of two components: a list of objects currently known to be in the scene and a set of matrices representing the primitive relations which have been found necessary and sufficient for performing further scene analysis.

Keeping a list of objects known to be in the scene allows the addition of further information to the Scene Model to be a trivial task. The object list component of the Scene Model is the set over which the primitive spatial (topological) relations is defined. Therefore adding the new tuples to the relations will only involve calculating the relations between the new entities and the new set over which the relations are defined. Thus the choice of dynamic model is feasible and will allow for a top-down scene analysis.

5.1. The Object List

The first component of the Scene Model is the object list. The objects on this list have been recognized during previous scene analysis operations end are represented by polyhedral surfaces alone. Thus to the high level reasoner it appears that objects are composed of only bounded planes - primitives at one level of representation. The use of a single primitive (or a set of primitives which are not composed from one another) is conceptually clean to work with and is adequate for modelling objects in this domain. Each instance of an object in the scene has the information associated with it which was determined necessary to facilitate further scene analysis. The components of an object record are a name, the list of faces comprising the object, its location in Euclidean three space, and an indication of the subtype of the object. The name of the object indicates the type of the object - one of the objects which can be expected in the current domain. The indication of the subtype gives more specific information about the object - the expectations one can have about an object when analyzing a scene. The face list represents the set of polyhedra which comprise the object. The metric location in Euclidean three space is approximated by the centroid of the object. The centroid of an object in this system is defined to be the average of all the centroids of the faces on the face list of that object.

5.2. The Relations

The relations in the Scene Model represent the primitive relations or topological properties between objects in the scene. The same six relations as in the Surface Model are adequate to represent all relations, both simple and complex, among objects using various forms of relational composition. These six relations - ADJACENT, CONTIGUOUS, LOOKSADJACENT, LOOKSCONTIGUOUS, ABOVE, and CONTAINS - are defined over the set of all objects currently recognized in the scene. The relations are represented by their adjacency matrices because the adjacency matrix is easily updated and makes composition of relations simple. The composition becomes a simple matter of boolean matrix multiplication for which there are many fast and efficient algorithms.

The definitions of the four relations are very similar to those of the Surface Model. CONTIGUOUS is a subset of ADJACENT. Two objects are said to be ADJACENT if they satisfy the following condition:

$\exists \text{FACE}_1 \in \text{OBJECT}_1$ and $\exists \text{FACE}_2 \in \text{OBJECT}_2$
such that FACE_1 ADJACENT FACE_2

OBJECT_1 is CONTIGUOUS to OBJECT_2 if:
$\exists \text{FACE}_1 \in \text{OBJECT}_1$ and $\exists \text{FACE}_2 \in \text{OBJECT}_2$
such that FACE_1 CONTIGUOUS FACE_2

Once again, CONTIGUOUS is a subset of ADJACENT. These two relations are only symmetric. LOOKSADJACENT and LOOKSCONTIGUOUS are defined similarly for the faces projected onto the x-y plane.

The ABOVE relation is computed by performing a simple comparison of the location fields, centroids, of the two objects. If the centroid of OBJECT_1 is higher from the ground than the centroid of OBJECT_2 then:
OBJECT_1 ABOVE OBJECT_2

The CONTAINS relation is the most difficult relation to compute . First the boundary of the face list component of each object must be calculated. These boundaries are then projected onto the xy-plane. The relation is then defined as follows:

$\text{Boundary}(\text{OBJECT}_1) \cap \text{Boundary}(\text{OBJECT}_2) = \text{Boundary}(\text{OBJECT}_1)$
then OBJECT_2 CONTAINS OBJECT_1
$\text{Boundary}(\text{OBJECT}_1) \cap \text{Boundary}(\text{OBJECT}_2) = \text{Boundary}(\text{OBJECT}_2)$
then OBJECT_1 CONTAINS OBJECT_2

ABOVE and CONTAINS are only transitive.

5.3. Justification of the Scene Model

It remains to show that the proposed Scene Model both adequately represents the objects in an image and facilitates the successful execution of analysis operations. As mentioned above, most of the analysis operations will fall into one of three categories: determining the existence of an object, finding an object part, or computing the relation which represents a locative construct relating objects. The existence of objects will be resolved in one of two ways - finding the object in the scene model by searching the object list, or using the recognizer to find a new instance of the object. To find a part of an object its face list will be searched until the part is found using the global knowledge about parts embodied in the object model. As for resolving the interpretation of locative constructs, the relations allow objects to be located relative to other objects in the scene using simple matrix operations. Suppose the question were asked, "Is there a car on the street?" An object of type CAR is ON an object of type STREET if the following primitive relations hold:

CONTAINS(STREET,CAR)
ABOVE(CAR,STREET)

The reasoner would determine if the CAR is ON the STREET by calculating the following relation composition:

$$\text{CONTAINS} * \text{ABOVE}^T$$

which would be calculated by a simple matrix multiplication of the CONTAINS adjacency matrix and the transpose of the ABOVE adjacency matrix. So the understanding of relational expressions will be accomplished by composing the primitive relations. The necessary compositions of primitive relations will be determined by linguistic knowledge used by the natural language interface in understanding the queries. Since the model facilitates these three operations which are essential to any scene analysis it, in fact, is robust enough to be used by the LandScan image understanding system.

6. Conclusion and Results

The recognizer and Scene Modeller are currently running on synthetic data. The low level modules have not yet been interfaced with the high level modules. However both have been tested on several different data sets. Figure 1 shows one synthesized data set with the surfaces labelled. Figure 2 shows some results of running the recognizer on this data.

This paper has described two modules of the LandScan system - the object recognizer and Scene Model. It is the Scene Model which is used by the natural language interface to answer queries about the scene. This representation of the image provides a module which allows the primitive spatial relations represented in this model to be combined and analyzed by the high level reasoning processes to reflect the meaning of the user's query. This provides a tool for analyzing a computational model for understanding locative phrases (natural language utterances about the spatial relations between objects).

The ATN formalism has been adopted for the recognition process. This choice was made because the formalism has a top-down flow of control which can be driven by natural language queries and a grammar which describes in a perspicuous way a method for finding the set of primitives which represents an object. The grammar does so by representing recognition as a generative process which finds a set of faces corresponding to a object in the scene. This search is constrained by the features which the faces must have and the various primitive relations which must hold between faces in order for the surfaces to be in the set defining the object. Although a recognition scheme in which primitives must be matched in a specific order seems an odd choice for visual processes, it has been shown that the formalism is applicable to the domain and at some future time, it might be possible to design an interpreter which is not constrained by a left-to-right parsing/generating strategy. An additional advantage of the ATN is that the recognition module will interface easily with high level reasoning processes. The reasoner will determine from the query the objects of interest. It will then call the recognizer and ask it to find those objects. When this has been done, the reasoner will be able to generate the proper response to the question. The recognizer also interfaces easily with the scene representation constructor using the BUILDQ action to add objects to the Scene Model.

A symbolic representation for the scene and objects in the scene has been presented which will facilitate high level reasoning processes driven by goal-oriented a· ·lysis. The dynamic Scene Model is constructed as LandScan is queried, thus reflecting the user's change in focus. The Scene Model has two components: a list of objects currently known to be in the scene and a set of primitive lccative relations between these objects. The object representation facilitates operations in which a part of an object is in question. The object list and recognizer will allow the existence of particular objects and object parts to be determined. It has been shown that the six primitive locative relations - ABOVE, CONTAINS, ADJACENT, CONTIGUOUS, LOOKSADJACENT, and LOOKSCONTIGUOUS - can be composed to obtain information about more complex relations between objects as embodied in locative constructs. Thus the recognition paradigm and Scene Model proposed will facilitate the top-down analysis of aerial images guided by natural language queries.

7. Acknowledgements

This research was supported by the following grants:

ARO DAA6-29-84-k-0061
AfOSR 82-NM-299
NSF MCS-8219196-CER
NSF MCS 82-07294
AVRO DAABO7-84-K-FO77
NIH 1-RO1-HL-29085-01
Also many thanks to Jon Korein for helping with SurfsUP.

Figure 1: Synthesized Urban Scene

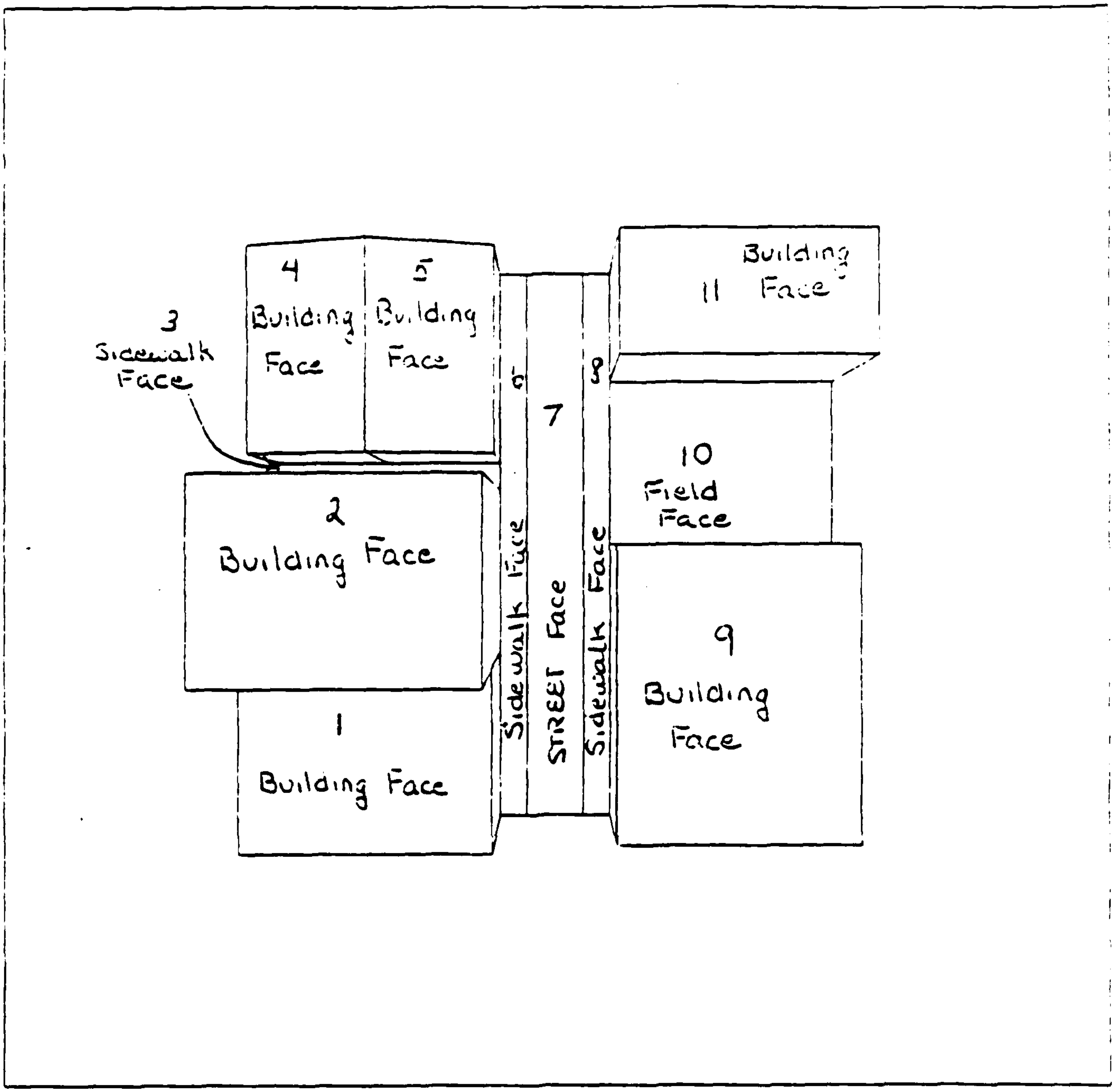


Figure 2: Results of Running Recognizer on Figure 1

```
LISP> (generate '[illegible])
OBJECT: [illegible]
SUBTYPE: COMPLEX
PIECES: 1

CONTIGUOUS
0

ADJACENT
0

LOOKSCONTIGUOUS
0

LOOKSADJACENT
0

ABOVE
0

CONTAINS
0

LISP> (generate 'sidewalk)
OBJECT: SIDEWALK
SUBTYPE: COMPLEX
PIECES: 3 5

CONTIGUOUS
0 0
0 0

ADJACENT
0 0
0 0

LOOKSCONTIGUOUS
0 1
1 0

LOOKSADJACENT
0 1
1 0

ABOVE
0 1
0 0

CONTAINS
0 0
0 0
```

```
LISP> (generate '[illegible])
OBJECT: [illegible]
SUBTYPE: [illegible]
PIECES: [illegible]

CONTIGUOUS
0 0 0
0 0 0
0 0 0

ADJACENT
0 0 0
0 0 0
0 0 0

LOOKSCONTIGUOUS
0 1 0
1 0 1
0 1 0

LOOKSADJACENT
0 1 0
1 0 1
0 1 0

ABOVE
0 1 1
0 0 0
0 0 0

CONTAINS
0 0 0
0 0 0
0 0 0

LISP> (generate 'street)
OBJECT: STREET
SUBTYPE: SIMPLE
PIECES: 7

CONTIGUOUS
0 0 0 0
0 0 0 1
0 0 0 0
0 1 0 0

ADJACENT
0 0 0 0
0 0 0 1
0 0 0 0
0 1 0 0
```

```
LOOKSCONTIGUOUS
0 1 0 0
1 0 1 1
0 1 0 0
0 1 0 0

LOOKSADJACENT
0 1 0 0
1 0 1 1
0 1 0 0
0 1 0 0

ABOVE
0 1 1 1
0 0 0 0
0 1 0 1
0 0 0 0

CONTAINS
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

References

[Bates 81] Bates, Madeleine. The Theoty and Practice of Augmented Transition Network Grammars. In Leonard Bolc (editor), *Natural Language Communication with Computers*. Springer-Verlag, 1981.

[Faugeras 81] Olivier D. Faugeras , Keith Price. Semantic Description of Aerial Images Using Stochastic Labeling. *IEEE Trans. on PAMI* PAMI-3(4):459-469, July, 1981.

[Fischler 83] Martin Fischler. *Image Understanding Research and its Application to Cartography and Computer-Based Analysis of Aerial Imagery.* Technical Report, SRI International, September, 1983.

[Fu 82] Fu,K.S. *Syntactic Pattern Recognition and its Applications.* Prentice-Hall, 1982.

[Glicksman 83] Glicksman, Jay. Using Multiple Information Sources in a Computational Vision System. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[Harlow 84] C. A. Harlow , R. W. Conners , M. Trivedi. A Computer Vision System for the Analysis of Aerial Scenes. In *Seventh International Conference on Pattern Recognition*, pages 407-410. July 30-August 2, 1984.

[Herman 83] Herman, Martin, Takeo Kanade, Shigeru Kuroe. The3D MOSAIC Scene Understanding System. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[Herskovits 82] Herskovits, Annette. *Space and the Prepositions in English: Regularities and Irregularities in a Complex Domain.* PhD thesis, Department of Linguistics, Stanford University, 1982.

[Herskovits 84] Herskovits, Annette. Space and the Prepositions in English: Regularities and Irregularities in a Complex Domain. 1984.Draft: University of California, Berkeley.

[Hwang 83] Hwang, Vincent, Takashi Matsuyama, Larry Davis, Azriel Rosenfeld. *Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding.* Technical Report, Center for Automation Research, University of Maryland, October, 1983.

[Krotkov 84] Krotkov, Eric. *Construction of a Three Dimensional Surface Model.* Technical Report, GRASP LAB, CIS Department, University of Pennsylvania, 1984.

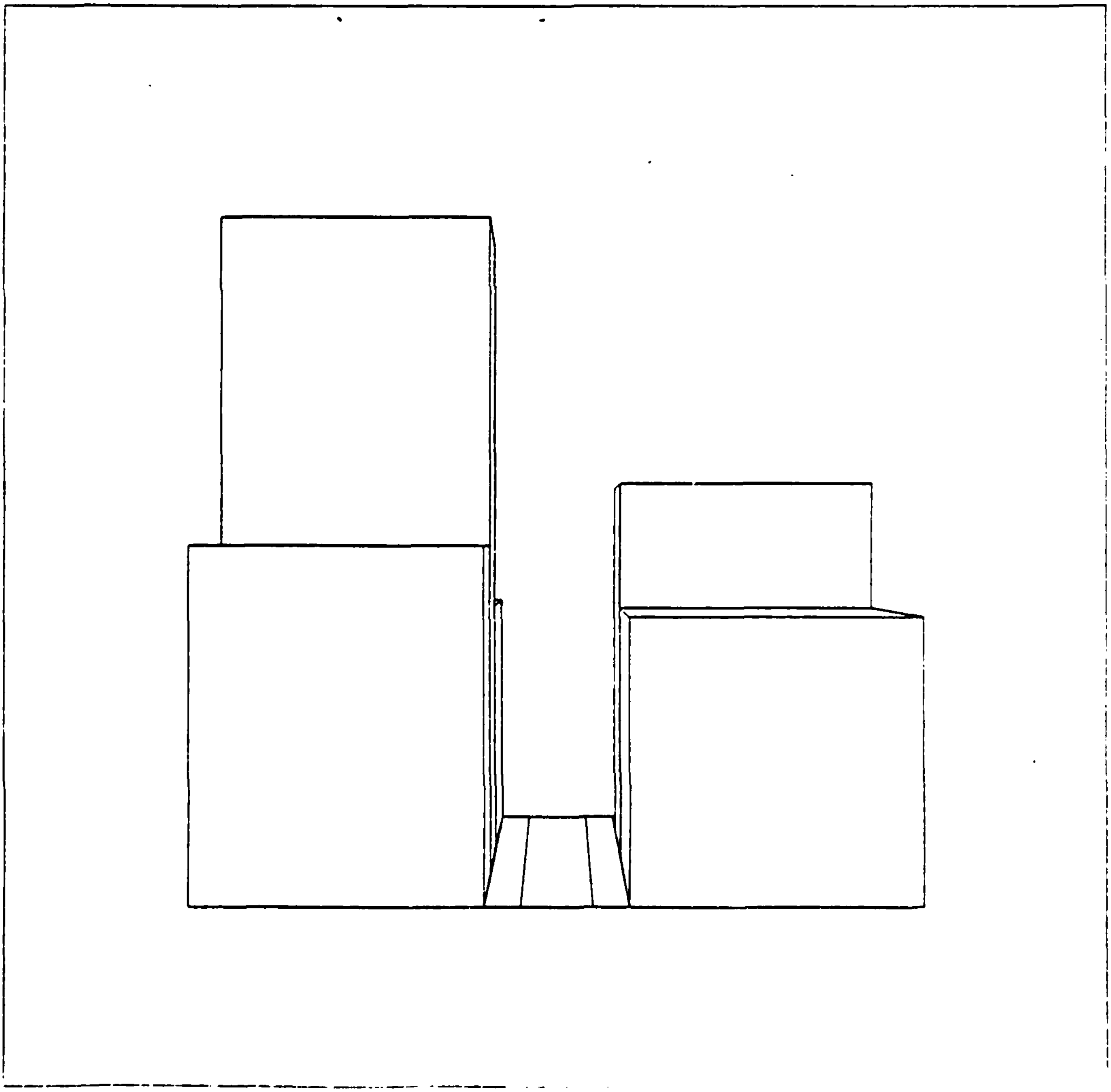
[Nagao 79] M. Nagao , T. Matsuyama , H. Mori. Structural Analysis of Complex Aerial Photographs. In *IJCAI 6*, pages 610-617. August 20-23, 1979.

[Nevatia 83] Ramakant Nevatia. *Final Technical Report.* Technical Report Report 104, Intelligent Systems Group, University of Southern California, October 19, 1983.

[Potmesil 83] Potmesil, Michael. Generating Models of Solid Objects by Matching 3D Surface Segments. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[Quam 78] L. H. Quam . *Road Tracking and Anomaly Detection in Aerial Imagery.* Technical Report Technical Note 158, SRI International, March 1978.

Figure 8-12: Reconstructed planar surfaces, rendered by Movie. BYU in side view.



References

[Akey 84] M. L. Akey , O. R. Mitchell. Detection and Sub-Pixel Location of Objects in Digitized Aerial Imagery. In *Seventh International Conference on Pattern Recognition*, pages 411-414. July 30-August 2, 1984.

[Canny 84] John F. Canny. *Finding Edges and Lines in Images.* Technical Report AI-TR-720, MIT, 1984.

[Faugeras 81] Olivier D. Faugeras , Keith Price. Semantic Description of Aerial Images Using Stochastic Labeling. *IEEE Trans. on PAMI* PAMI-3(4):459-469, July, 1981.

[Fischler 83] Martin Fischler. *Image Understanding Research and its Application to Cartography and Computer-Based Analysis of Aerial Imagery.* Technical Report, SRI International, September, 1983.

[Grimson 81] W.E.L. Grimson. *From Images to Surface: A Computational Study of the Human Early Vision System.* MIT Press, 1981.

[Harlow 84] C. A. Harlow , R. W. Conners , M. Trivedi. A Computer Vision System for the Analysis of Aerial Scenes. In *Seventh International Conference on Pattern Recognition*, pages 407-410. July 30-August 2, 1984.

[Herman 83] Herman, Martin, Takeo Kanade, Shigeru Kuroe. The3D MOSAIC Scene Understanding System. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[Hwang 83] Hwang, Vincent, Takashi Matsuyama, Larry Davis, Azriel Rosenfeld. *Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding.* Technical Report, Center for Automation Research, University of Maryland, October, 1983.

[Izaguirre 84] Alberto Izaguirre , Pearl Pu , John Summers. *A New Development in Camera Calibration -- Calibrating a Pair of Mobile Cameras.* Technical Report MS-CIS-84-55, University of Pennsylvania, 1984.

[Krotkov 84] Krotkov, Eric. *Construction of a Three Dimensional Surface Model.* Technical Report, GRASP LAB, CIS Department, University of Pennsylvania, 1984.

[Lee 85] Yong Hoon Lee, Saleem A. Kassam. Generalized Median Filtering and Related Nonlinear Filtering Techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33, accepted for publication in 1985.

[Marr 80] David Marr, Ellen Hildreth. Theory of Edge Detection. In *Proc. R. Soc. Lond.*, pages 187-217. 1980.

[Nagao 79] M. Nagao , T. Matsuyama , H. Mori. Structural Analysis of Complex Aerial Photographs. In *IJCAI 6*, pages 610-617. August 20-23, 1979.

[Nevatia 83] Ramakant Nevatia. *Final Technical Report.* Technical Report Report 104, Intelligent Systems Group, University of Southern California, October 19, 1983.

[Quam 78] L. H. Quam . *Road Tracking and Anomaly Detection in Aerial Imagery.* Technical Report Technical Note 158, SRI International, March 1978.

STEREO PROCESSING OF AERIAL IMAGES

Rusena Bajcsy
David L. Smitley

MS-CIS-84-36
GRASP LAB 07

Department of Computer and Information Science
Moore School/D2
University of Pennsylvania
Philadelphia, PA 19104

December 1983

[*] Acknowledgement: This research was supported in part by NSF grants MCS8219196-CER, MCS-82-07294, 1 RO1-HL-29985-01, U.S. Army grants DAA6-29-84-K-0061, DAABO7-84-K-FO77, U.S. Air Force grant 82-NM-299, AI Center grants NSF-MCS-83-05221, US Army Research office grant ARO-DAA29-84-9-0027, Lord Corporation, RCA, and Digital Equipment Corporation.

Stereo Processing of Aerial Images

David L. Smitley & Ruzena Bajcsy

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104

Introduction

Over the years much work in computer vision has been done using aerial images as input data. Typical applications involving analysis of aerial images are automatic map generation, analysis of natural resources, analysis of reconnaissance photos, and guidance systems for autonomous dying vehicles. Currently, we are working on a complete system that will take as input a high resolution, low altitude stereo pairs of urban scenes and will produce a high level description of the scene. It is hoped that a query language can be formulated with which inquiries such as, "How tall is the building under the cursor?" can be answered.

To achieve this goal 3-D information must be extracted from the image. To extract 3-D information we have opted to use passive stereo. The main advantages of this technique is that it passive (as opposed to active techniques such as radar) and it requires no special range gathering hardware other than the cameras themselves.

While stereo techniques have been successful in the field of cartography, their usefulness has been limited when used to extract 3-D information from urban scenes. The major problem with stereo is that of determining the correspondence between pixels in the two images. A pixel in one image may have many candidate correspondences in the other.

In urban scenes the correspondence problem is made even more difficult, do to several factors. A major problem is brought about by the way the images are obtained. To create a large enough baseline for adequate stereo separation the aircraft that is taking the photos must first take one photo, travel a distance equal to the baseline and take another. Two problems arise from such a process. One is the existence of moving objects such as automobiles.

The matcher must now distinguish between disparities caused by motion and disparities caused by depth. Another problem created by this difference in time and position is that there can exit photometric differences between the two images. Atmospheric variations such as clouds can cause the two images to differ in focus, contrast, and average intensity, hence making uniform feature extraction difficult.

In addition to the problems introduced by having two images taken at differing times, the rapid changes in depth that are prevalent in urban scenes also increase the difficulty of the correspondence problem. Rapid changes in depth cause occlusions. Thus certain features visible in one image may not be visible in the other. The matcher must therefore take into account the fact that not all features in one image need match a feature in the other image.

Goals

We wish to generate a relatively accurate depth map given two aerial, stereo photos of an urban scene. Here we are working with images of Washington, D.C. We will not attempt to generate a complete disparity map using only stereo. That is, we will not try to match every pixel in the two images. One reason that we are not attempting to generate a complete map is the feeling that a complete, accurate depth map would be impossible to obtain from the images we are providing as input. Second, we feel that if we are provided with accurate depth points in locations that lend themselves to reasonable interpolation, then we can do an adequate job of filling in the depth map during a 3-D interpretation phase, possibly aided by monocular depth cues. Another requirement of the depth map generator is the need that it be formulated in a way that parallel implementation using hardware could be realized. In addition, we would like the matcher to be as robust as possible thereby extending its applicability beyond that of analyzing aerial photos of cities.

A review of the literature leads to the classifying of stereo matching techniques into two major groups, that of area based matching and the other group that attempts to match features such as edges, or zero crossings of an image convolved with a difference of gaussian

(DOG) operator.

The area based approach attempts to match the two images by taking a window from one image and finding what part of the other image gives a maximum correlation. Since correlation is mathematically expensive some methods use an "interest operator" to select those areas of the image that are to be matched [Moravec 1980], [Gennery 1980]. The area based approach has been shown to produce good results when processing natural scenes but when applied to cultural scenes with many occluded surfaces the results are unacceptable. Thus, we decided to use feature based matching as the basis for our depth map generator.

Several algorithms based on feature matching seemed to be particularly applicable to aerial image processing. One such method developed by Grimson [Grimson 1980] matches features which have been extracted by sensing the zero crossings in the difference of two Gaussians or as it is more commonly referred to in the literature, the DOG filter. Furthermore, the image is processed at four different spatial resolutions with the matches from the lower resolution filters guiding the matching process at the higher resolutions.

An algorithm developed by Baker [Baker 1982] and a similar one developed by Arnold [Arnold 1983] also seemed to be appropriate to the task of generating the results we desired. While there exist some differences between the algorithms, the main principle behind both of them is that of achieving an optimum set of matches for each epipolar line in the image. The way that this optimum match is found is by assigning each possible way of combining the edges found in each line a probability that one matches with the other. This probability is calculated from numerous parameters are gathered from the edge and other local information from the neighborhood around the edge. The assignment of edge correspondences which produce the largest total probability is then computed using a dynamic programming technique known as the Viterbi algorithm [Forney 1973].

Method

As mentioned previously, we decided to match features along epipolar lines. The current version of the matcher uses zero crossings of an image convolved with the DOG operator [Grimson 1980]. However, before extracting the features for matching we filter the images using a non-linear filter which we call a "double window smoother" [Lee 1983]. The filter consists of both a median filter in combination with a mean filter. However, the two are joined in such a manner that almost all impulse noise is removed, high frequency noise is suppressed and edges are still retained. While the filter is effective in removing noise, its effects on the extraction of zero crossings is minimal, hence the process adds to the robustness of the matcher.

As in Grimson's matcher we filter the image with a dog filter:

$$\left(\frac{r^2-2\sigma^2}{\sigma^4}\right)e^{-\frac{r^2}{2\sigma^2}}$$

$$w = 2\sqrt{2}\sigma$$

at four different widths (w = 32, 16, 8, 4) and use the matches from the lower resolution images (large w) to guide the matching process at higher resolutions (small w). However, the matching process at each level and the method by which the previously found matches are used to guide higher resolution matching are significantly different from Grimson's method.

At each resolution level a two step process is used to determine whether a given zero crossing in one image corresponds with a zero crossing in the other image. I will call the two phases absolute and probabilistic. In the absolute phase several characteristics of the two zero crossings in question are compared and if they fail to meet a certain criteria the correspondence probability is a given a value of zero. If the pairing is not rejected in the absolute phase a probability is calculated that the two edges match. These probabilities (from both the absolute and probabilistic phase) are then entered into a matrix, and the modified Viterbi algorithm [Baker 1982] is used to determine the disparity profile that produces the largest total probability.

Absolute Phase

If an edge pairing fails to meet any of the following criteria a probability of zero is assigned to the edge.

C1. The difference between the orientation of the two zero crossings must be less than 30 degrees.

C2. The signs of the slopes of both zero crossings must be the same.

C3. The disparity must be less than some a priori determined maximum disparity.

C4. If a zero crossing in the current resolution image is within +-w/2 of a matched zero crossing in the previous image, the disparity must be within +-w/2 of the disparity assigned to the zero crossing in the previous image.

C5. All disparities must be of the same sign (monotonicity).

C6. The search range for corresponding zero crossings is constrained by the matches found in the previous resolution image. Any corresponding zero crossings found outside this range are rejected.

C1 thru C3 require no further discussion as they intuitively seem reasonable, and have been shown to be effective disambiguators by Grimson. C4 was chosen as heuristic based on the assumption that the previous matches were correct and therefore a match in neighborhood around this match in a higher resolution image should have a similar disparity. The localization error of the zero crossings appears to be +-w/2. That is, a zero crossing is within +-w/2 pixels of the actual intensity change giving rise to the zero crossing. Hence, the choice of a neighborhood size of +-w/2.

A major assumption was made about the images. It is assumed that the images possess the property of montonicity. Monotonicity is the property that all disparities are of the same sign. That is if:

L_1 *matches with* R_1 *and* $L_1 > L_2$

then

L_2 *must match with an an* R_2 *such that*

$R_2 > R_1$

where L_n *is the y coor. of zero crossing in left image*

and R_n *is the y coor. in right image.*

Tall, narrow objects or large overhangs lead to a violation of this assumption. However, the near orthographic projection found in aerial images severly limits the chances that such a violation would occur. Assuming monotonicity and accurate previous matches, allows one to use C5 and C6 as absolute criteria. Thus, C5 enforces monotonicity in the current resolution matches. If C6 is violated, monotonicity between the current resolution matches and the matches from the previous level is violated. Assuming monotonicity also allows one to use the Viterbi algorithm to determine the maximum probability disparity profile.

Probabilistic Phase

After the absolute phase there can still exist more than one candidate match for a given zero crossing. Each of these remaining pairings is assigned a probability based on what I will call one-sided correlation.

As was mentioned earlier, area based matching was successful in creating disparity maps for scenes with small amounts of occlusion. However, they failed when presented scenes with a large number of occluding surfaces. The reason for failure is that when trying to correlate an area of an image which contains an occluding surface, part of that area has no correlate. Thus, the correlation found will be useless for that part of the image. In addition, correlation is computationally extensive. One sided correlation overcomes these problems.

In matching, only those zero crossings with orientations greater than 30 degrees from horizontal are considered since disparity information from horizontal edges is extremely difficult to achieve. In one-sided correlation a correlation window is centered around the zero crossing and is divided into two regions by a vertical line. A normalized correlation value is then calculated for both the left and right hand regions formed by the vertical line, using the following equation:

$$\sigma(q_n) = [E(q_n^2) - (E(q_n))^2]^{\frac{1}{2}}$$

$$N = \frac{E(q_1 q_2) - E(q_1)E(q_2)}{\sigma(q_1)\sigma(q_2)}$$

Normalized correlation is used to help eliminate the effects of photometric differences between the two images. The probability that the two zero crossings match is then assigned to the maximum correlation value of the two regions. If this probability is less than some threshold (chosen to be .85) the probability is set to zero.

The above procedure eliminates the occlusion problem associated with normal correlation. To see this consider the following possible results from one-sided correlation:

Case 1. Correlation values from both left and right hand side of edge are low.

Case 2. Correlation values from both left and right hand side of edge are high.

Case 3. Correlation value for left hand side of edge is low, and right hand side is high.

Case 4. Correlation value for right hand side of edge is low, and left hand side is high.

Case 1 corresponds to the case where the given zero crossing correspondence is incorrect. Case 2 corresponds to the situation where the zero crossings are NOT occluding thus both areas on the left and right sides of the zero crossing cause high correlation values. Cases 3 and 4 are similar in the sense that they arise when the zero crossings in question are occluding. Thus, the area on one side of the zero crossing need not correlate but the area on other side (if this is indeed a correct pairing) will. Hence, in any one of the cases a high correlation value for at least one side of the zero crossing corresponds to an actual match.

The forementioned process thus handles the occlusion problem. It also deals with the computational expense of correlation, as only the most likely pairings have a correlation value calculated for them.

After the correlation values are calculated for each zero crossing pair and thresholded, there still remains ambiguous matches. Therefore, the probabilities for all pairings are entered into a matrix and the modified Viterbi algorithm described by [Baker 1982] is used to determine the most likely disparity profile. The pairings giving rise to this profile are then recorded and used to guide the next higher resolution matching process. As these matches are required to guide the next level of matching, a simple linear interpolation process is performed to give disparity values to all vertical neighbors of zero crossings that have been matched thereby increasing matching performance at the next lowest level.

Results

Matching results for one image (figure 1) can be seen in figures 2 thru 5. These are the results from the various resolution filters (w = 32, 16, 8, 4). The bottom image in each figure are the zero crossings matched from the left image and shifted by their corresponding amount of disparity. In addition, the intensity of a pixel is directly proportional to its disparity. The number of incorrectly matched pixels can be estimated from this display by comparing pixel intensities to those of their neighbors. Any sharp difference represents inconsistent matches. Furthermore, if the shifted pixels vary greatly in position from those of the right image, this too represents incorrect matches. Using these factors as criteria, the number of incorrectly matched zero crossings in the highest resolution (w = 4) appears to be quite small (less than 10). The number of zero crossings in the left image is 2264 hence only .5% of the matches are incorrect. While the number of incorrect matches is low, the number of accurate matches is enough to allow the generation of a surface using simple linear interpolation between matched zero crossings on a given line. If there does not exist two zero crossings to interpolate between, the disparity values are obtained from the previous line. This technique is currently used only to produce a displayable disparity map and a more robust interpolation technique may be needed to produce a more accurate, complete disparity map from the sparse match points. Figure 6 is an isometric view of the surface generated using this technique.

Conclusions

One-sided correlation in combination with some simple, absolute criteria for matching produces many accurate matches. Further improvements in the quality of matches could possibly be achieved if some other feature than zero crossings were used. One might consider using a robust edge detector such as the one proposed by Canny. Another area for possible improvement lies in the interpolation of the surface. An accurate disparity map generated at each level would help the next lower level of matching reject false matches.

One-sided correlation appears to disambiguate false matches as accurately as standard correlation but avoids the problems that arise when attempting to deal with occlusions. This feature is highly desirable when working with aerial images of urban scenes where occlusion is prominent. In addition one-sided correlation can be used to determine whether a matched zero crossing is an occluding zero crossing [Witkin]. This additional information can the be used to guide the interpolation step since knowing whether an edge is occluding can greatly constrain possible surfaces.

Furthermore, the speed of our algorithm could be greatly increased with the addition of special purpose hardware. High speed processors to perform the DOG and zero crossing operations have already been developed [Nisha 1983]. In addition, since correlation is a well understood and widely used mathematical technique, many high performance algorithms and processors exist to do correlation at speeds greater than those achievable on a typical SISD computer. Since no inter-line dependencies exist, a third method for achieving greater speed is to dedicate a processor per line.

Several proposed algorithms [Baker 1982][Ohta 1982] for the stereo matching problem attempt to increase the accuracy of the matches by using three dimensional consistency between matches as a constraint. While this technique is effective at removing a large number of false matches, the final result usually retains a small percentage of false matches. Hence, one needs to weigh the advantages of reduced false matches (but not all false matches) to the disadvantage of the increased processing required to enforce three dimensional consistency. If the matching process produces a very small number of false matches (as does one-sided correlation) one should consider dealing with the false matches at interpolation time as opposed to trying to excise every false match (a near impossible task) with such techniques as three dimensional consistency.

Finally, the need to use dynamic programming to provide the optimum disparity profile can be questioned. Usually, no ambiguous matches are entered into the matrix and seldom does a given zero crossing have more than two candidate matches. Thus, a simpler method of disambiguating such as simply taking the maximum probability of the ambiguous match to be the match, might be as effective as dynamic programming in building the final profile. However, a useable procedure has yet to be developed.

In conclusion one-sided correlation appears to be an effective, efficient means of extracting depth information from a stereo pair.

[Arnold 1983] Arnold, R. David, "Automated Stereo Perception," Department of Computer Science, Stanford University, Ph.D. thesis, 1983.

[Baker 1982] Baker, H. Harlyn, "Depth from Intensity and Edge Based Stereo," Department of Computer Science, Stanford University, 1982.

[Ballard 1982] Ballard, Dana H., and Christopher M. Brown, "Computer Vision," Prentice Hall, Englewood Cliffs, N.J., 1982.

[Forney 1973] Forney, G. David Jr., "The Viterbi Algorithm," Proc. IEEE, Vol 61, No. 3, March 1973, 268-278.

[Gennery 1980] Gennery, Donald B., "Modeling the Environment of an Exploring Vehicle by Means of Stereo Vision," Ph.D thesis, Stanford Artificial Intelligence Laboratory, June 1980.

[Grimson 1980] Grimson, William Eric Leifur, "From Images to Surfaces: A computational Study of the Human Early Vision System," MIT Press, 1981.

[Hildreth 1980] Hildreth, E., "Implementation of a Theory of Edge Detection," MIT Artificial Intelligence Memo 579, April 1980.

[Lee 1983]. Lee, Yong, Personal communication, paper forthcoming. University of Pennsylvania, Moore School of Electrical Engineering, 1983.

[Marr 1982] Marr, David, "Vision," W.H. Freeman and Co., San Francisco, CA, 1982.

[Marr 1978] Marr, D. and Hildreth, E., "Theory of Edge Detection," MIT Artificial Intelligence Memo 518, April 1978.

[Marr 1977] Marr, D. and T. Poggio, "A Theory of Human Stereo Vision," MIT Artificial Intelligence Memo No. 451, November 1977.

[Marr 1976] Marr, D. and T. Poggio, "Cooperative Computation of Stereo Disparity," Science, Vol. 194, October 1976, 283-287.

[Moravec 1980] Moravec, Hans p., "Obstacle Avoidance and Navigation in the Real Worl by a Seeing Robot Rover," Stanford Artificial Intelligence Laboratory, Ph.D. thesis, September 1980.

[Nisha 1983] Nishihara, H.K., "PRISM: A Real-Time Imaging Stereo Matcher" Proceedings of the Third International Conference on Robot Vison and Sensory Controls", Society of

Photo-Optical Instrumentaion Engineers, 1983.

[Ohata 1983] Ohata, Y. and Kanade T., "Stereo by Intra- and Inter-scanline Search Using Dynamic Programming" Carnegie-Mellon University, Memo # CMU-CS-83-162, October 1983.

[Talton 1983] Talton, David, Personal communication, paper forthcoming. University of Pennsylvania, Department of CIS, 1983

[Witkin] Witkin, Arnold P., "Intensity Based Edge Classification," Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, CA.

Figure 1. Windows selected for matching

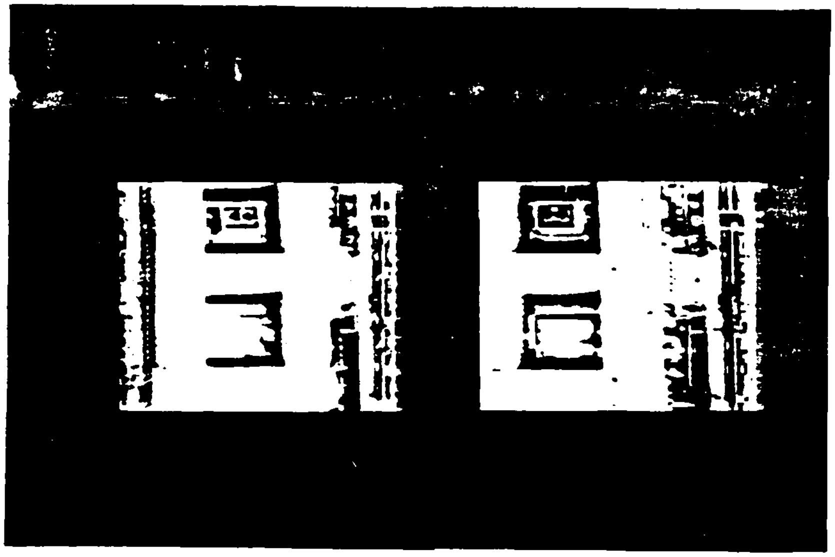


Figure 2. Results with filter width = 32.
Top left image - edgels detected in left image
Top right image - edgels detected in right image
Bottom image - edgels matched in left image shifted and weighted by their disparity.

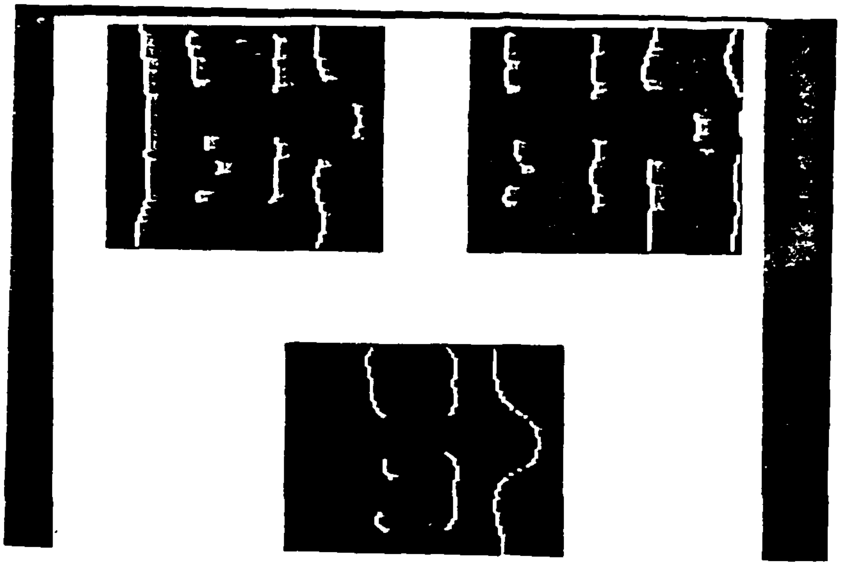


Figure 3. Results with filter width = 16.
Top left image - edgels detected in left image
Top right image - edgels detected in right image
Bottom image - edgels matched in left image shifted and weighted by their disparity.

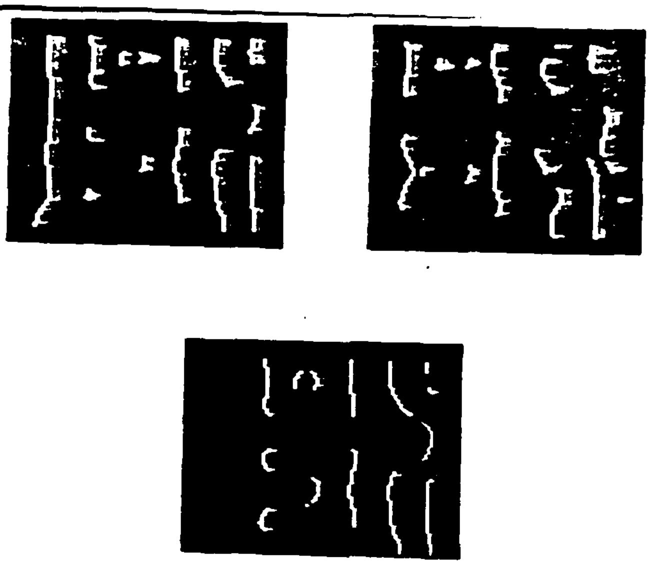


Figure 4. Results with filter width = 8.
Top left image - edgels detected in left image
Top right image - edgels detected in right image
Bottom image - edgels matched in left image shifted and weighted by their disparity.

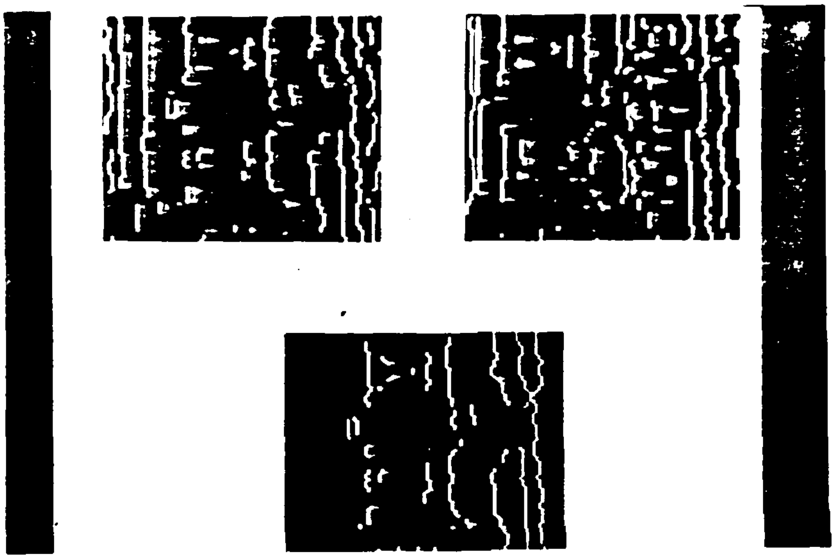


Figure 5. Results with filter width = 4.
Top left image - edgels detected in left image
Top right image - edgels detected in right image
Bottom image - edgels matched in left image shifted and weighted by their disparity.

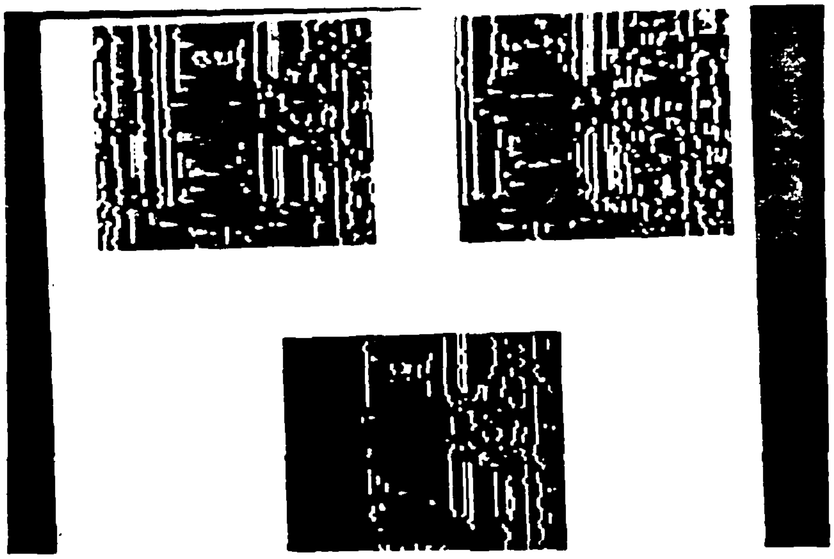
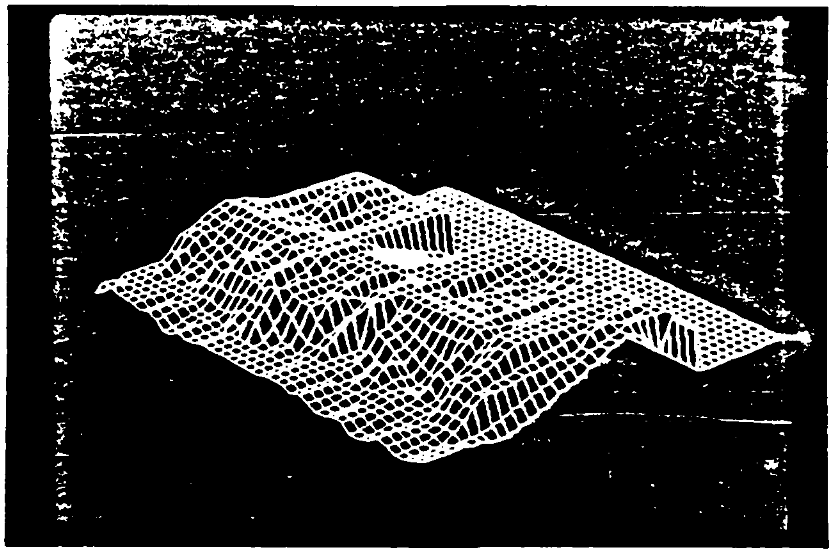


Figure 6. Isometric plot of interpolated uisparities



IMPLEMENTATION OF A GAUSSIAN-SMOOTHING GRADIENT-BASED EDGE DETECTOR

By David Talton

University of Pennsylvania
GRASP Lab Memo
MS-CIS-85-XX

Abstract

This report describes the theoretical aspects and implementation details of a gaussian-smoothing, gradient-based edge detector. This edge detector is based on Canny's "Finding Edges and Lines in Images" [1]. In this report we discuss the implementation of an algorithm and the results rather than the motivation for the computation.

This report describes research done at the GRASP Laboratory of the University of Pennsylvania's Moore School of Electrical Engineering. Support for the laboratory's vision research is provided in part through the following contracts:

ARO DAA6-29-84-k-0061
AfOSR 82-NM-299
NSF MCS-8219196-CER
NSF MCS 82-07294
AVRO DAAB07-84-K-FO77
NIH 1-RO1-HL-29985-01

[Radack, et al 84]
Radack, Korein, Ganis, McNally, Korein, Shapiro. *NASA Programmer's Guide* CIS Department, University of Pennsylvania, 1984.

[Reynolds, et al 84]
Reynolds, G, N.Irwin, A.Hanson, E.Riseman. Hierarchical Knowledge-Directed Object Extranction Using a Combined Region and Line Representation. In *Vision Workshop*. 1984.

[Rosenthal 81] Rosenthal, David. *An Inquiry Driven Vision System Based on Visual and Conceptual Hierarchies*. UMI Research Press, 1981.

[Shapiro 84] Shapiro and Haralick. A Heirarchical Relational Model for Automated Inspection Tasks. In *Int. Conf. on Robotics, Atlanta, Ga.*. 1984.

[Sloan 81] Kenneth R. Sloan , P. G. Selfridge. Reasoning about images: Applications to aerial image understanding. In *Proc. 1981 Image Understanding Workshop*, pages 1-6. April 1981.

[Talmy 83] Talmy, Leonard. *How Language Structures Space*. Technical Report 4, Berkeley Cognitive Science Report, January, 1983.

[Tropf 83] Tropf and Walters. An ATN for 3-D Recognition of Solids in Single Images. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. 1983.

[Winograd 83] Winograd, Terry. *Language as a Cognitive Process*. Addison-Wesley Publishing Co., 1983.

[Winston 79] Winston, Patrick Henry. *Artificial Intelligence*. Addison-Wesley Publishing Company, 1979.

This report describes the implementation of an edge detection algorithm based on the computation of the gaussian smoothed gradient of an image. It describes the computation of the gradient and the procedure for locating edges (Canny[1]). The ideas for this algorithm are from Canny[1]. This algorithm locates local-maxima in the gaussian smoothed gradient of the image at a particular scale. No attempt is made to combine results across scales. The two steps are described below: image gradient computation and non-maxima suppression.

IMAGE GRADIENT COMPUTATION

Let $I(x,y)$ be the image intensity function and
$G(x,y)$ be a gaussian filter where

$$G(x,y)=e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

We wish to compute the gradient of the gaussian smoothed image at some scale, this is

$$\nabla f=\nabla(G*I)$$

where

$$f(x,y)=G(x,y)*I(x,y)$$

The scale is determined by the standard deviation (σ) of the gaussian filter.

Now,

$$\nabla f=\mathrm{i}f_x+jf_y$$

or,

$$\begin{aligned}\nabla f&=\nabla(G*I)\\&=i(G*I)_x+j(G*I)_y\\&=i(G_x*I)+j(G_y*I)\end{aligned}$$

This means that to compute the image gradient we compute its components: the x-directional derivative and y-directional derivative of the image.

From above we have,

$$f_x = G_x * I$$

$$f_y = G_y * I$$

and the filters are,

$$G_x(x,y) = \frac{-x}{\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

$$G_y(x,y) = \frac{-y}{\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

Thus, we compute the directional derivatives of the image at a particular scale by convolving the image with the filters $G_x(x,y)$ and $G_y(x,y)$. From the directional derivatives we compute the gradient magnitude as

$$\nabla f = \sqrt{f_x^2 + f_y^2}$$

and the gradient direction as

$$DIR = \tan^{-1}\left\{\frac{f_y}{f_x}\right\}$$

The filters $G_x(x,y)$ and $G_y(x,y)$ above, are 2-dimensional filters. Because these are separable filters we may compute the above convolutions by convolving twice with one dimensional filters.

This is,

$$G_x(x,y) = G_x(x) * G(y)$$

$$G_y(x,y) = G_y(y) * G(x)$$

and the computation of f_x and f_y becomes

$$f_x = G_x(x) * G(y) * I$$

$$f_y = G_y(y) * G(x) * I$$

The order of the convolutions does not matter. To compute the gaussian smoothed gradient of the image four one-dimensional convolutions are needed. The filter coefficients are computed by integration of the filter over the pixel area rather than simple sampling. As usual [1][5][6], we can vary the scale of the gradient calculation by varying the standard deviation (σ) of the gaussian (low-pass) filters.

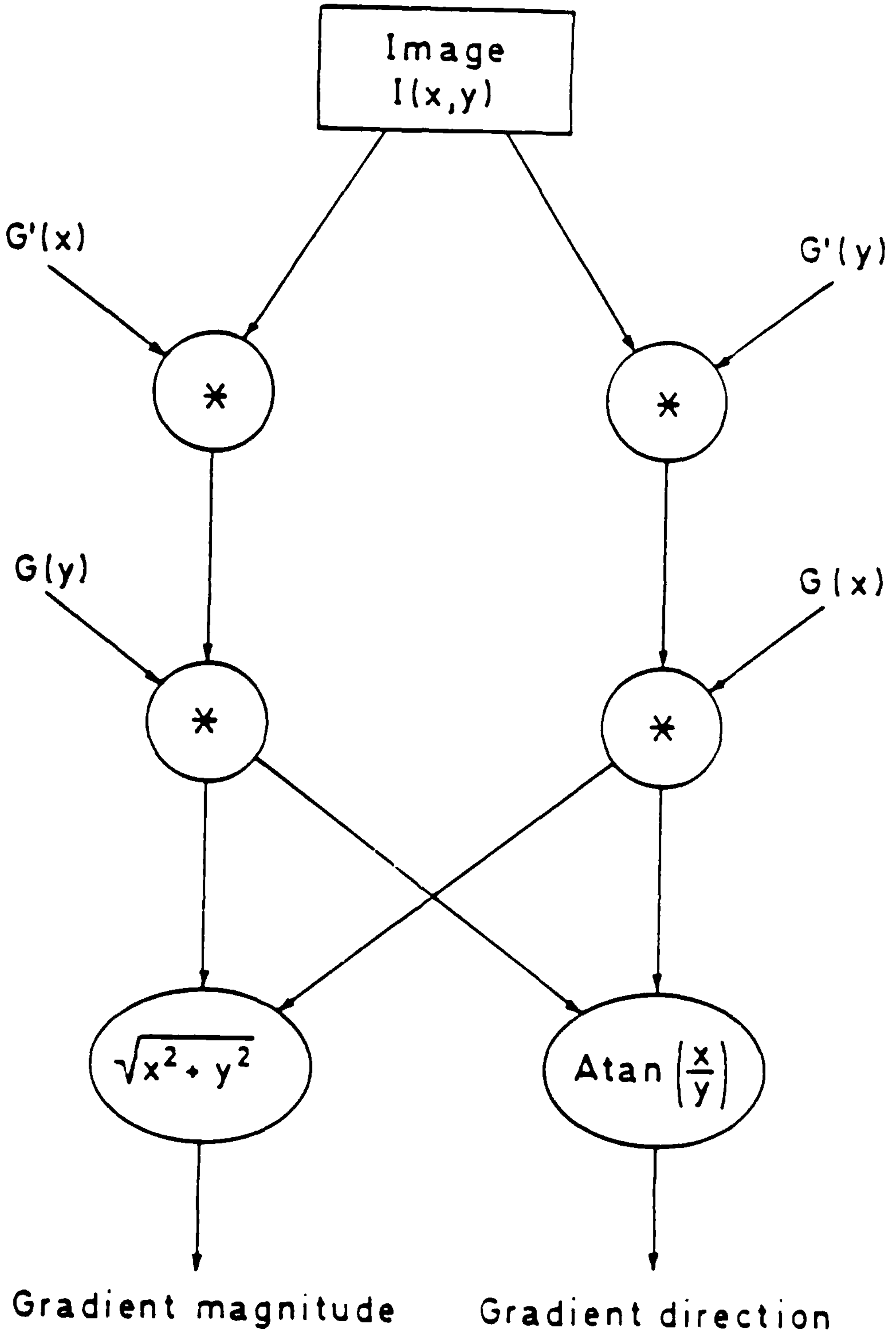
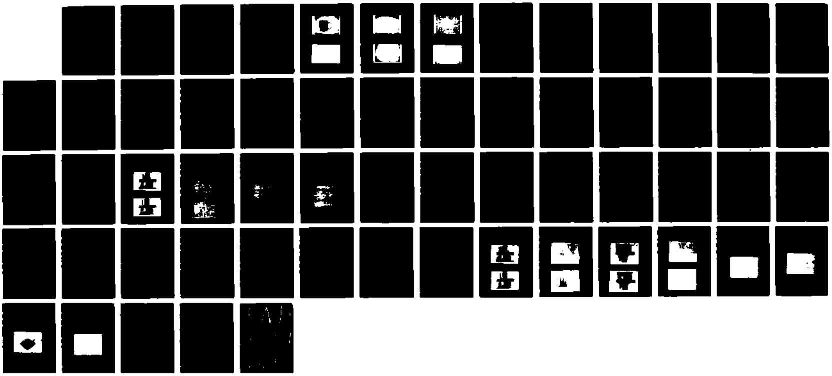


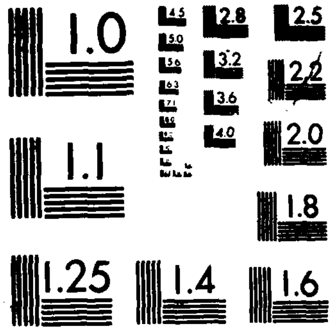
Image
I(x,y)
G'(x)
G'(y)
*
*
G(y)
G(x)
*
*
$\sqrt{x^2+y^2}$
Atan $\left|\frac{x}{y}\right|$
Gradient magnitude
Gradient direction

$\nabla(G * I)$

DATA FLOW DIAGRAM

AD-A185 687 A QUERY DRIVEN COMPUTER VISION SYSTEM: A PARADIGM FOR 2/2
HIERARCHICAL CONTRO..(U) PENNSYLVANIA UNIV PHILADELPHIA
SCHOOL OF ENGINEERING AND APPL.. R BAJESKY APR 83
UNCLASSIFIED AFOSR-TR-87-1205 F49620-83-K-0037 F/G 17/5 NL




1.0
4.5
5.0
5.6
6.3
7.1
8.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

NON-MAXIMA SUPPRESSION

To locate local maxima in the image gradient we use Canny's scheme for non-maxima suppression [1]. That is, we compare the gradient magnitude for a given pixel (A) with the interpolated gradient magnitudes in the gradient direction (at points B and C).

Non-maxima suppression [1].

If the gradient magnitude at point A *is greater than* the magnitude at both points B and C, point A is designated a local maxima. Computationally, it is easiest to make this comparison using the directional derivatives f_x and f_y before computing the gradient vectors because the interpolation weights are ratios of these values (see Canny[1] p. 82-83). See figure 3.

CROWLEY'S "PEAKNESS"

Another method for locating maxima in the gradient magnitude array is Crowley's [3] "peakness" measure. This method compares a pixel's gradient magnitude with the gradient magnitude of 8 neighboring pixels. Pixels which lie on ridges have a high "peakness" because they have a higher gradient magnitude than most of their neighbors. The edge maps in figures 4 and 6 were generated using Crowley's method.

PERFORMANCE EVALUATION

How can we judge the performance of an edge detector? Just what does it mean to detect edges? What is the purpose of edge detection? There is little agreement in the vision community over the definition of an edge and (I believe) a growing concern over the purpose of edge detection and the nature of low-level vision. Edge detection has usually been used as a preprocessing step for some higher-level vision task (i.e. stereo, shape, etc.). Researchers have been overly concerned with edge-detection methods, to the point of ignoring the edge detector's interaction with other visual processes. The procedure for edge detection discussed above also ignores interaction with other processes. But, since we have developed a fairly successful adaptation of the edge detection described in Canny[1], I will discuss the more direct aspects of this implementation.

This implementation consists (as most edge detectors do) of two fundamental steps: derivative computation and feature detection. To be more specific, this edge detector fits the mold of a "detection function" (Crowley[2]). That is, linear filtering, followed by a non-linear decision procedure. The algorithm described above also adds an in-between step (non-linear, filter result combination) because we are computing the gradient magnitude. I am most satisfied with the method of derivative computation, and less satisfied with the non-linear decision procedure called "non-maxima suppression". There are several ways to detect ridges in the gradient, and we intend to study a few of them quantitatively. These include Canny's[1] non-maxima suppression, Crowley's ridge detector[3] applied to the gradient magnitude map and another method of my own device based on Haralick's Topographic Primal Sketch[4].

The non-maxima suppression scheme of Canny is sensitive to scale in that the distance to the comparison points should vary with resolution. It is not obvious for which scale the non-maxima suppression scheme is best. In general detection functions (which work on a 9-pixel neighborhood) used after smoothing with gaussians of different sigmas are not equivalent.

We seek connected, smooth, 1-pixel wide contours. Canny's non-maxima suppression scheme does not insure this. I am interested in a local procedure where nearby edges reenforce each other over the one-to-two pixel range to produce short edge segments. I am interested in producing this directly from the decision procedure.

The method described for computing the linear, gaussian-smoothed, directional derivatives of the image is directly generalizable to the computation of the second derivatives f_{xx}, f_{yy} and f_{xy}. These are computable by separable filters as above. I have seen no implementation of the laplacian of a gaussian [5] using these (separable) derivative filters. All implementations use the Difference of Gaussians as an approximation to the laplacian of a gaussian because the DOG can be implemented using separable gaussian filters. But implementing the laplacian of a gaussian can be done directly using the filters specified above. The computational complexity is the same as the difference of gaussians. (four one-dimensional convolutions)

In the future we will investigate the combination of edge detection results across scales[6]. We have not addressed the question of sampling frequency in scale space (one octave, half octave or something else), this again will be left for future study.

Figure 1. Cup image.

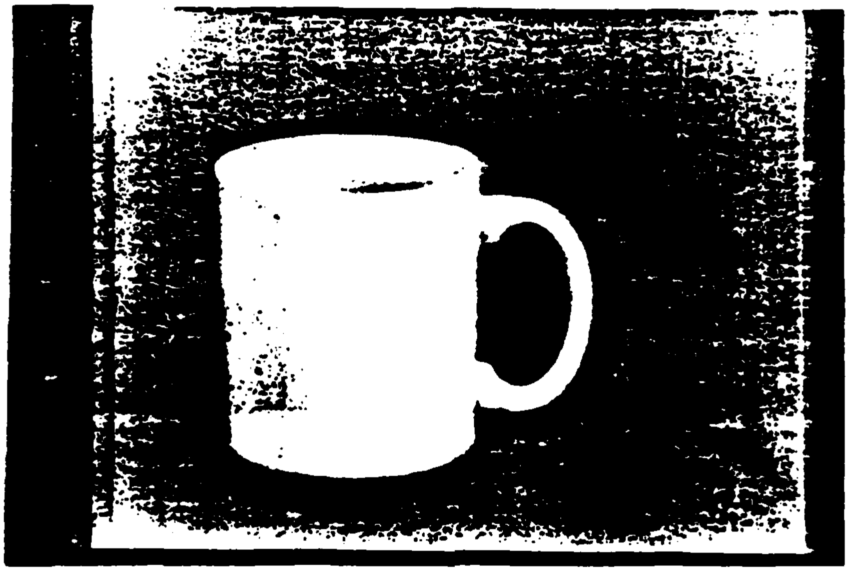


Figure 2. Gaussian-smoothed gradient magnitude of image.

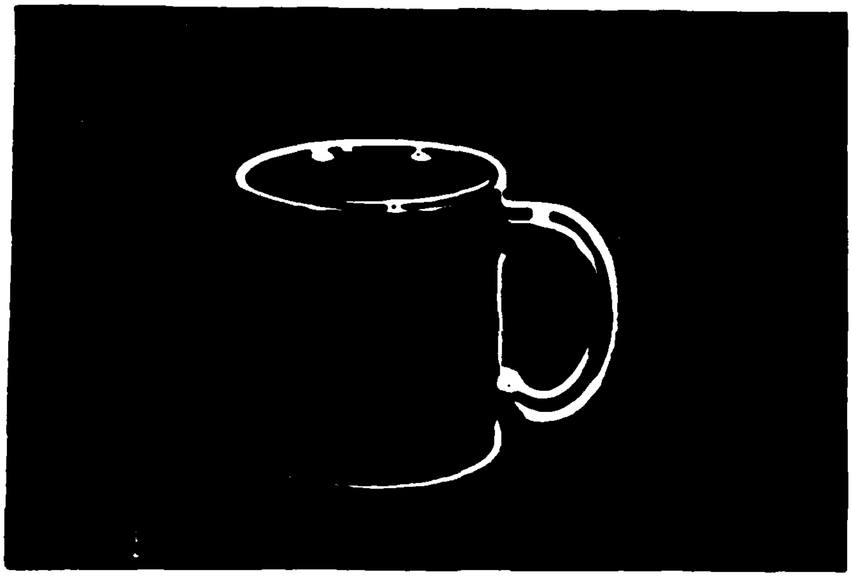


Figure 3. Edges detected using Canny's [1] non-maxima suppression.

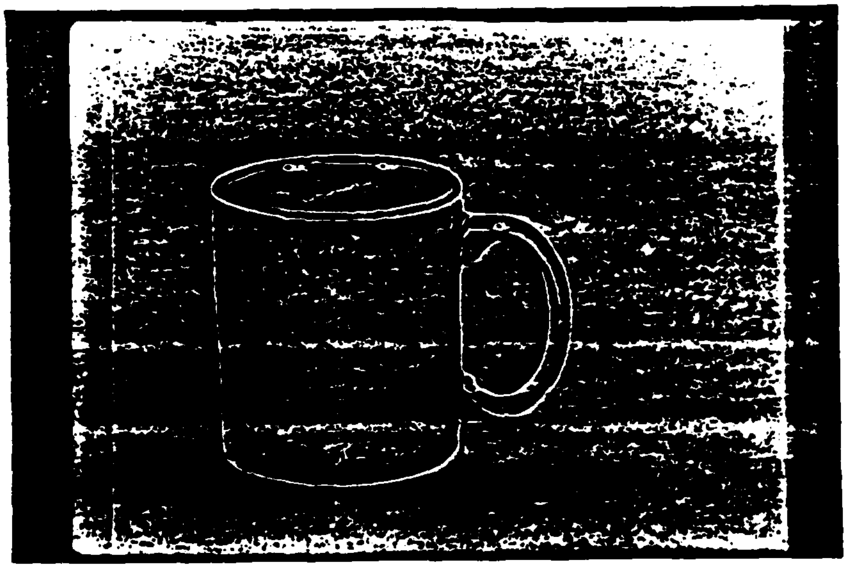


Figure 4. Edges detected using Crowley's [3] ridge detector.

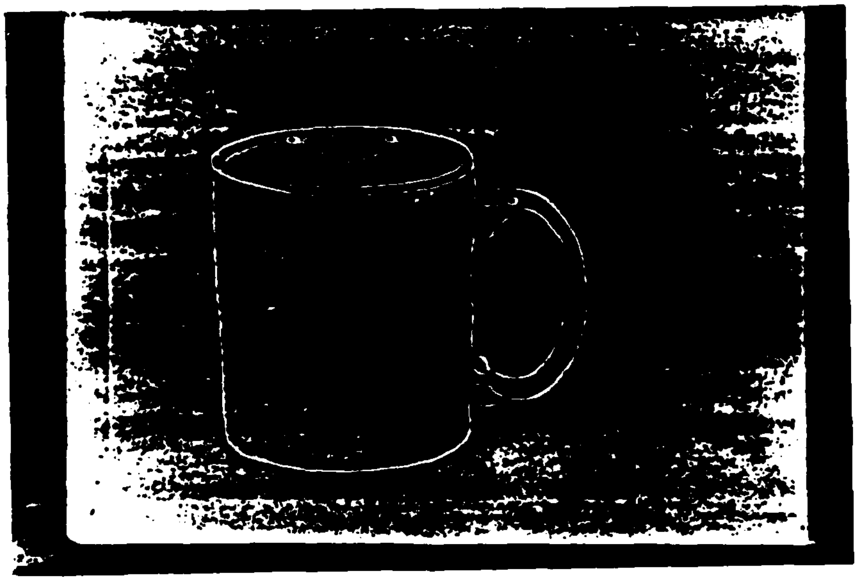


Figure 5. Washington D.C. image.

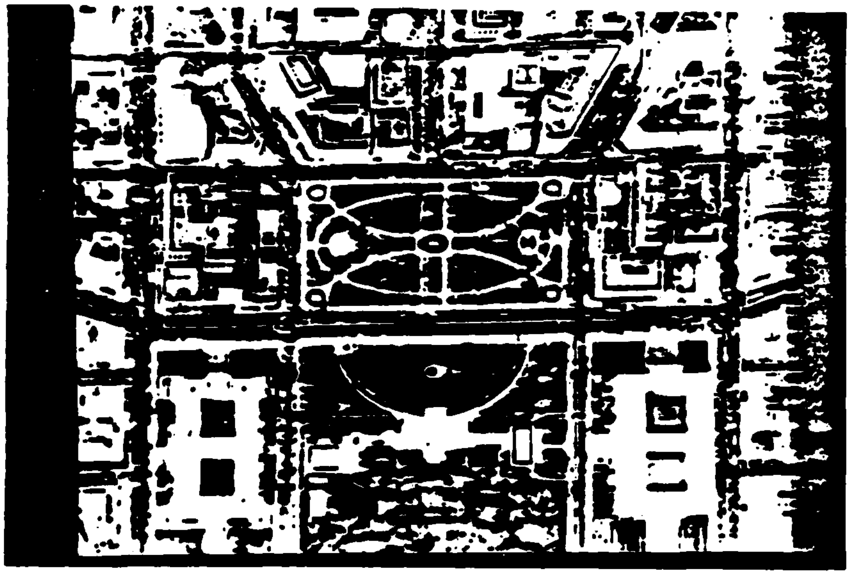
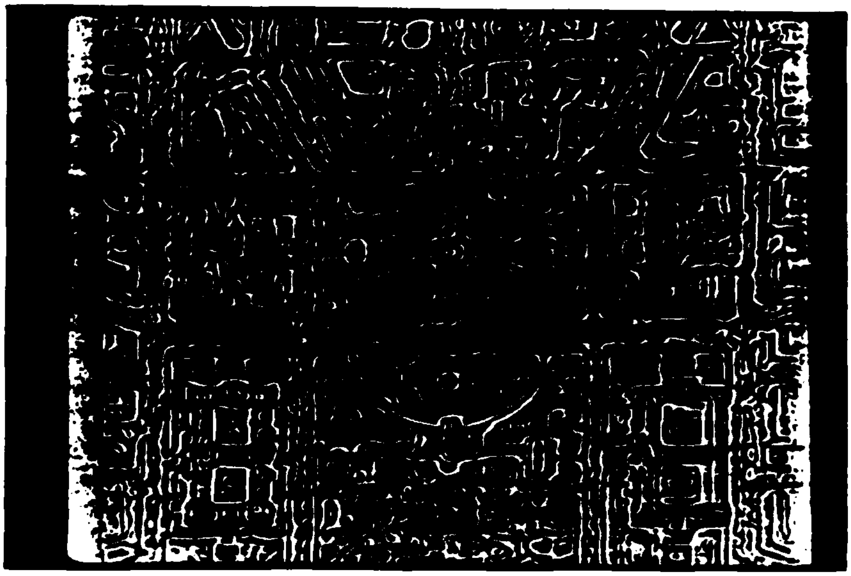


Figure 6. Edges detected using Crowley's ridge detector on the gaussian smoothed gradient magnitude.



REFERENCES

[1] Canny, John Francis, "Finding Edges and Lines in Images", MIT AI Memo 720, June 1983.

[2] Crowley, James, "A Representation for Visual Information",CMU-RI-TR-82-7, November, 1981.

[3] Crowley, James, "A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform",CMU-RI-TR-83-4, May, 1983.

[4] Haralick, Watson and Laffey, "The Topographic Primal Sketch", International Journal of Robotics Research, Vol. 2,No. 1, 1983.

[5] Hildreth, E.C., "Implementation of a Theory of Edge Detection", MIT AI Memo 579, April 1980.

[6] Witkin, Andrew, "Scale Space Filtering: A New Approach To Multi-Scale Description", Image Understanding 1983.

LandScan: A Natural Language and Computer Vision System for Analyzing Aerial Images

Ruzena Bajcsy
Aravind Joshi
Eric Krotkov
Amy Zwarico

CIS Dept/D2
University of Pennsylvania
Philadelphia, Pennsylvania 19104
(215) 898 - 3191

topic: Natural Language and Perception
keywords: vision, natural language, integrated system
length: 5461 words (not counting references)

Abstract

LandScan (LANguage Driven SCene ANalysis) is presented as an integrated vision system which covers most levels of both vision and natural language processing. Computations are both data-driven and query-driven. In this report we focus on the design of the vision modules. Future work will investigate in more detail the design of the natural language interface.

The data-driven system employs active control of stereo cameras for image acquisition, and the bottom-up flow of control dynamically constructs a surface model from multiple aerial views of an urban scene.

The query-driven system allows the user's natural language queries to focus analysis to pertinent regions of the scene. This is different than many image understanding systems which present a symbolic description of the entire scene regardless of what portions of that picture are actually of interest.

A top-down flow of control dynamically generates a scene model after creating the surface model. The object recognizer is an ATN in which the grammar describes sequences of primitives which define objects and the interpreter generates these sets of primitives. The scene model is dynamically built as objects are recognized, representing both the objects in the image and primitive spatial relations between these objects.

Abstract

LandScan (LANguage Driven SCene ANalysis) is presented as an integrated vision system which covers most levels of both vision and natural language processing. Computations are both data-driven and query-driven. In the report we focus on the design of the vision modules. Future work will investigate in more detail the design of the natural language interface.

The data-driven system employs active control of stereo cameras for image acquisition, and the bottom-up flow of control dynamically constructs a surface model from multiple aerial views of an urban scene.

The query-driven system allows the user's natural language queries to focus analysis to pertinent regions of the scene. This is different than many image understanding systems which present a symbolic description of the entire scene regardless of what portions of that picture are actually of interest.

A top-down flow of control dynamically generates a scene model after creating the surface model. The object recognizer is an ATN in which the grammar describes sequences of primitives which define objects and the interpreter generates these sets of primitives. The scene model is dynamically built as objects are recognized, representing both the objects in the image and primitive spatial relations between these objects.

LandScan:
A Natural Language and Computer Vision System for Analyzing Aerial Images

1. Introduction

The aim of our research on LandScan (LANguage Driven SCene ANalysis) is to develop a system capable of dynamically updating and maintaining a model of an urban world over multiple aerial views. The system will have a natural language front end, through which users can query the system about what it sees, and to direct or interactively assist the vision processing by restricting the analysis to those areas of the scene which are of current interest, dynamically constructing models as the system is queried.

A unique contribution of the work is that processing is both data-driven ("bottom up," determined by sensor data) and query-driven ("top down," determined by user queries). The integration of both methods into one system can help overcome the shortcomings of each method employed independently. For example, if data-driven processing were able to segment a graph of edges derived frcm the image into *several* different connected components, query-driven information about what the system *should* be looking for can help impose structure, and a unique segmentation, upon the otherwise ambiguous data.

The data-driven processing starts with stereo aerial images and proceeds, by filtering, matching, interpolating, and fitting, to reconstruct the surfaces in the scene. The aerial domain buys the simplicity of *planar* surfaces. Two factors distinguish this data-driven system from many others. First, image acquisition is controlled by feedback from the query-driven system, and is undertaken by active sensors, actively probing the environment. Second, the controlled environment of a scale urban model is a testbed allowing precise verification of results and individual modules (it is being tested on real images as well).

For query-driven processing an Augmented Transition Network (ATN) has been chosen to perform the object recognition because it has a top-down flow of control thus facilitating the interface between the queries and the recognition process. The scene will be represented symbolically by the objects which have been recognized and the primitive spatial relations which hold between them. The Linguistic Analyzer performs syntactic analysis of the query to produce a symbolic representation which is then processed by th Reasoner. The Reasoner, using global knowledge of the domain will perform the following reasoning operations:

1. determining the existence of an object
2. finding an object part
3. determining locative relations, both simple and complex, among objects.

It will also handle in a non *ad hoc* manner query failure. Also, the state of the Scene Model represents the history of the user's interest in the scene.

This paper will describe the aerial domain, some related research, the implementation of the data-driven and query-driven portions of the LandScan system, and our plans for future work. A later paper will detail how natural language queries will interface with LandScan to guide the scene analysis.

2. The Aerial Domain

Aerial images suffer from a poverty of context due to the great imaging distances. Urban scenes contain featureless areas and large numbers of occlusion edges. Even with the best possible use of image data, we generally can do no better than to compute a sparse depth map of the imaged scene. For many purposes a sparse depth map is inadequate, and the missing surface information must be obtained from other sources: other "shape from ..." processes, domain-dependent high-level knowledge, and real-world constraints.

There are two major constraints in the aerial domain:

1. The data is obtained by taking *aerial* photographs of an *urban* environment. Urban scenes are characterized by an abundance of straight lines. This means that to a very good approximation the scene, as viewed from on high, is composed of planar polyhedra, so that detected edges separate planar surfaces, i.e., each edge arises because it is the intersection of 2 planar faces.

2. The image acquisition process is under our control, so the camera model is known. Some combination of azimuth and elevation angles, Euler angles, pan. roll, ti'lt angles are available and fully specify a 4x4 homogeneous transformation relating the position and orientation of the two cameras.

Domain knowledge includes such facts as roofs of buildings tend to be parallel to the ground plane, while walls are perpendicular to it, and that sidewalks are thinner (more compact) than roads.

3. Related Research

A large corpus of research on aerial image understanding *per se* exists, [akey84], [harlow84], [Hwang83], [nagao79], [sloan81], [quam78], [faugeras81], [Glicksman83], [reynolds84], [potmesil83], and many general vision techniques are applicable to the aerial domain. Large aerial projects have been undertaken at USC [nevatia83] and at SRI [Fischler83].

The 3D MOSAIC project [Herman83] is geared toward the urban aerial domain. Important differences in their strategy are that junctions are primitive, and a monocular analysis is performed. At the level of object representation LandScan treats surfaces as primitive, while 3D Mosaic treats faces, edges and points as primitives. The LandScan representation of objects by their surface primitives was chosen because it is compact, easy to analyze, and a representation sufficient for matching. Further, in 3D MOSAIC hypotheses are generated about the continuation of occluded lines, shapes of faces, and the extent of vertical faces. The construction of the scene model in the 3D MOSIAC system is exclusively data-driven, while LandScan uses a query-driven approach for constructing the Scene Model.

ATN's have been used primarily in the domain of natural language [Bates81], [winograd83], [winston77]. A notable exception to the use of ATN grammars for natural language understanding is the system designed by Tropf and Walter [tropf83] which uses an ATN model for the recognition of 3D objects with known geometries. The recognition process performed by their system is one of "analysis-by-synthesis" in which hypothetical model instantiations about an object (prototypes) are generated and then verified by the ATN. These prototypes are then verified by comparing them to the actual 3D data using the ATN. If the similarity between the prototype and the image exceed some threshold then the prototype is considered to be a model instantiation of the actual data. Otherwise, another prototype is generated and matched against the data.

Shapiro and Haralick [Shapiro84] describe a hierarchical, relational 3D model which is influential in our design. Their model provides precise, accurate information to be used by low-level vision and inspection processes as well as information required by high-level vision and reasoning processes. All of the information is represented by using "spatial data structures", each consisting of a recursive set of relations. The hierarchy consists of four levels: world, object, part, and surface/arc.

Rosenthal [Rosenthal81] proposed a model and interpreter for analyzing aerial images of urban settings. In some ways, Rosenthal's work was the impetus for this system. He proposed a purely hierarchical model of the world which is ordered by the ON relations and a goal driven production system to control recognition. It has a database which contains descriptions of objects and regions. He introduced an Object Description Notation to describe objects in the scene. This notation contains information about both the actual and possible properties.

The work of Talmy and Herskovits [Talmy83] [Herskovits82] [Herskovits84] has influenced the design of both the topological relations in the models and the choice of linguistic attributes which must be associated with objects in order to insure a robust and reliable natural language interface. It is from their work that the need for a single meaning for a single relation was discovered. Herskovits methodically discusses the knowledge we as natural language users have about the objects which we use in spatial constructions every day.

4. data-driven System Implementation and Results

This section will describe the implementation and performance of the major modules Figure F1: image acquisition, image enhancement, edge detection, stereo matching, mapping disparity to depth, interpolation of sparse depth points, edge segment fitting, construction of surface graph, surface attributes and topological relations between surfaces.

We show results derived from imaging a scale model. We have also tested the modules on real, highly complex aerial images.

4.1. Image Acquisition

Presently images are acquired manually by positioning cameras above a scale model of some toy buildings. Figure F2 illustrates a typical stereo pair of images acquired. A system for automatically setting camera parameters (location, pan, tilt, focus, zoom, aperture, vergence angle) has been constructed, and a controller for optimizing the parameters on the basis of feedback from high-level goals, medium-level strategies, and low-level image features is under design.

This "smart camera" is an *active sensor*, capable of moving in or out for a better look, zooming in on a feature, improving its signal/noise ratio,.and "craning its neck" for a better vantage. Our philosophy is to have the sensors do as much of their own processing as possible in a heterarchical environment, and not to devote all our resources to exhaustive analysis of a static scene.

4.2. Image Enhancement

Before extracting the features for matching the images are smoothed with a non-linear double window median filter [Lee85], removing impulse noise, and suppressing high-frequency noise. Independently, the range of grey scales is extended to cover 256 values by linear contrast enhancement.

4.3. Edge Detection

Two edge operators have been implemented, the Canny [Canny84], [Talton84] and DOG zero-crossing [marr80] methods. As implemented, these operators return "edgels," defined as points possibly lying at an intensity discontinuity, rather than "edges," defined as a set of edgels lying along a space curve. The two operators are now being carefully evaluated and compared on the basis of false positives, false negatives, and overall robustness under focus degradation and illumination degradation. Although the verdict is not yet in, the Canny operator is presently employed, and typical results are shown in Figure F3.

The Canny operator approximates a directional first derivative. The direction information can be used to find areas of high curvature (e.g., corners). Our present approach is to look at the variance of the directions in a small edgel neighborhood to identify corners and junctions.

4.4. Stereo Matching

Because of the large interocular distance in the aerial (fly-by) imaging there are large disparity jumps and large portions of the scene are visible in one image but not the other. This occlusion problem has haunted many matchers.

The matcher [Smitley84] employs the method of 2-sided correlation in order to circumvent some of the difficult problems of occlusion, and uses a registration technique to bring the scan lines into correspondence [Izaguirre84]. Figure F4 illustrates the results from the matcher. Present work in matching concerns evaluating its robustness, extension to higher-order features (e.g., linear segments, corners, and junctions), and obtaining horizontal disparities as well by taking three views per stereo frame instead of two.

4.5. From Disparity to Depth

Generally both disparity (distance in image space between matching pixels) and depth (distance in 3-space from viewer to object) are measured in a viewer-centered coordinate system. The function from disparity to depth (absolute, not relative) is linear in the disparity, interocular distance, focal length, and vergence angle. In the case where the view vector is parallel to the ground, a large disparity implies that the object is close, i.e., has a *small* depth value. In the case where the view vector is perpendicular to the ground (i.e., in the aerial domain) a large disparity implies that the object is close, i.e.. is far from the ground. We adopt the convention of mapping large disparities into large depths.

The method is essentially triangulation. We are building hardware to both control and measure the vergence angle between two cameras. With this angle, the exact distance to any point fixated in both visual fields can be discovered. Given this exact distance, the relative depth map returned from stereo can now be fixed as an absolute depth map.

4.6. Depth Point Interpolation--Filling In The Gaps

Presently two types of interpolation are implemented. The first attempts to restore edgels which should have been matched, but were not matched, by comparing the depth map with a map of edgels with a largely vertical (hence matchable) component in its directional derivative. The depth map is updated by adding selected edgels with a linearly interpolated depth value. This is an important process, and the results of linear intepolation are not entirely satisfactory. Improved interpolation will use corners and junctions in the near future.

The second interpolation fills depth values in featureless areas. This is quite simple-minded [grimson81] and does not provide exceptional results. But because this is used primarily for display purposes, i.e., we do not want to hypothesize about featureless areas, this is not a significant problem.

4.7. Edge Segment Fitting--Generating Wire Frames

This process fits a set of (straight) line segments in 2-point form (wire frames) from a rich set of depth points by a divide-and-conquer method of recursive decomposition. This method assumes that the boundary is of low curvature, and needs information about the location of corners to operate correctly. Figure F5 illustrates the edge segments generated from an interpolated depth map, and corners specified interactively.

4.8. Surface Model

A graph is constructed to serve as the surface model. This process converts a set of contours into a set of *closed* contours represented as a graph (a linked list of vertices, edges, and faces). The construction algorithm looks for minimum distance paths from a vertex back to itself, by traversing edges and at trihedral junctions choosing the path making the most acute angle with respect to the present path. Figure F6 illustrates the faces represented in the surface model.

Surface attributes and relations are computed in the SurfsUP [Radack84] geometrical modeling system. In it, a *face* is defined by its enclosing 3D contours. Attribute values for each face in the surface graph are computed [Krotkov84]: compactness, centroid vector, (outward-pointing) normal vector, area, "type," (building, sidewalk, field, street, and unknown), and number of sides. These values are computed once and stored on an attribute list.

Computed topological relations are *above*, *adjacent* (touching), *contiguous* (sharing an edge), *contains* (proper inclusion), *looksadjacent*, *lookscontiguous* (respectively adjacent and contiguous under perspective transformations) [Krotkov84]. Relations (and indirectly their complements) are computed once and stored as Boolean arrays. These relations are expensive to compute because they require intersection operations (except the *above* relation).

5. query-driven System Implementation and Results

This section describes the design and implementation of the query-driven processes. These include object recognition, scene modelling, high level reasoning processes, and query handling.

5.1. Object Recognition

The Augmented Transition Network (ATN) formalism has been chosen as the paradigm for object recognition in LandScan. It is composed of three parts: the grammar, a dictionary, and an interpreter. The grammar represents the *a priori* or world knowledge that the system must have in order to recognize objects and assign "cultural" labels to subsets of the scene. The dictionary is simply a list of all of the faces which have been segmented by the low and middle level routines. It represents the actual data which will be used in the recognition process. The third component of the recognizer is the Lisp program which provides the control structure for the process. An object is recognized by traversing a network successfully. Figure F7 shows the results of running the recognizer on the image in Figure F6.

The ATN formalism was chosen to perform object recognition for several reasons. First, the grammar enables the global knowledge about object appearances to be encoded as a generative model (grammar) for objects of indefinite appearances. Another reason is that the ATN operates using a top-down control structure - enabling the object recognition to be a query-driven process. Finally, the fact that the global knowledge (grammar) and the control structure are separate makes adding more global knowledge or changing the control strategy trivial.

The grammar as written is a two level network (this is considerably simpler than most ATN's which handle natural language utterances.) The bottom level concerns itself with the recognition of "simple objects." An object is simple if its further decomposition into parts will result in no entity which is in the domain of objects. For example, decomposing a building with a pitched roof will result in two halves of a pitched roof. Neither of these entities are considered objects in the domain - they are parts of objects. This level consists of the networks SIMPBUILD, SIMPSTREET, SIMPFIELD, and SIMPSIDEWALK. The top level combines the simple objects which were recognized in the first level of the network into "complex objects". A complex object is decomposable in a nontrivial way into at least one simple object.

A network is a set of nodes and arcs. The nodes represent how far the system has progressed in the object recognition. The arcs represent the patterns (object primitives of simple objects) which must be matched in order to proceed further in the recognition of that particular object. Each network is represented by a set of grammar rules.

The states are named with the convention of two part names [bates81]. The first part of the name indicates the name of the network and the second part describes either how far along this state is in the recognition process or the subtype of the object being recognized.

The arcs are represented by lists of the form (TYPE HEAD TEST ACTION). TYPE indicates the category or type of arc. The arc types in LandScan are:

- PUSH - call to a "simple" network
- CAT - search the dictionary (surface model) for an appropreate face

- POP - return to a calling network or add an object to the scene model and return that an object has been found
- JUMP - go to the next state without searching for a primitive object or face

HEAD is dependent upon the type of arc it is. HEAD can be a syntactic category - words or lists of words, a constituent type, the next state, or the form in which the data "parsed" is to be returned. TEST is a list (possibly empty) of tests to be performed before the arc can be traversed. The tests on the arcs encode the relations which must hold betweeen the components of an object and also provide further checking of the features of a component. ACTION is a list of actions to be performed as the arc is traversed. The possible egister setting and structure building actions are:

- (SETR REG VALUE) - sets the register REG to the evaluation of VALUE
- (SETRQ REG STRING) - sets the register REG to the literal STRING
- (ADDR REG VALUE) - appends the evaluation of VALUE to the end of the list in REG
- (BUILDQ <OBJECT>) - builds an object instance

There are two registers associated with the system - a SUBTYPE register and an OBJECT register. The SUBTYPE register contains the current subtype of the object being recognized. It is a feature register whose value is a string indicating the subtype name of an object. The OBJECT register is a role register containing a list of all the faces which comprise the object. As faces are found which match the generative sequence described by the grammar, they are added to the OBJECT register.

The dictionary in the recognizer is the Surface Model (described above) which represents both the geometric and topological information about the surface primitives in the scene [krotkov84].

5.2. The Scene Model

In order to perform any scene analysis in a reasonable way the scene has to be represented in some fashion which will enable the operations necessary for scene analysis to be performed. These actions include determining the relations, both complex and simple, among objects; and locating and identifying specific objects and object parts. A dynamic scene model has been designed which is composed of two components: a list of objects currently known to be in the scene and a set of matrices representing the primitive relations which have been found necessary and sufficient for performing further scene analysis. The scene model is dynamic because information can be added to it as further image analysis occurs.

The first component of the scene model is the object list. The objects on this list are those objects which have been recognized during previous scene analysis operations. These objects are represented only by polyhedral surfaces. Each instance of an object in the scene has the information associated with it which was determined necessary to facilitate further scene analysis. The components of an object record are a name, the list of faces (polyhedral surfaces) comprising the object, its location in Euclidean three space (average of the centroids of all the faces comprising the object), and a subtype which gives more specific information about the expectations one can have about the object.

The relations in the scene model represent the primitive relations or topological properties between objects in the scene. The six relations adequate to represent all relations, both simple and complex, among objects using various forms of relational composition are ADJACENT, CONTIGUOUS, LOOKSADJACENT, LOOKSCONTIGUOUS, ABOVE, and CONTAINS. They are defined over the set of all objects currently recognized in the scene. These relations are defined similarly to their counterparts in the Surface Model. The relations are represented by their adjacency matrices because the adjacency matrix is easily updated and makes composition of relations simple. The composition becomes a simple matter of boolean matrix multiplication for which there are many fast and efficient algorithms.

Keeping a list of objects known to be in the scene allows the addition of further information to the scene model to be a trivial task. The object list component of the scene model is the set over which the primitive spatial (topological) relations is defined. Therefore adding the new tuples to the relations will only involve calculating the relations between the new entities and the new set over which the relations are defined. Thus the choice of dynamic model is feasible and will allow for a top-down scene analysis.

5.3. Linguistic Analyzer

Given a query, the Linguistic Analyzer will symbolically represent this utterance so that it can be used by the reasoning process to analyze the image. The Linguistic Analyzer will parse the query, determine the query type, and categorize all implicit subqueries in the actual utterance. The output from the analyzer will contain a list of the objects to be found, the relations which must hold between these objects, and the query type (so that an appropriate response can be generated). As an example of this query analysis, suppose the user were to ask the question, "Is there a car on the street?" The output from this query would be the objects to be recognized, car and street; the relation ON defined be multiplying the CONTAINS relation by the transpose of the ABOVE adjacency matrix; and an indication that this query is responded to by a yes/no answer with some explanation. In this phase, the analyzer may discover that the query fails to have an answer because the query is syntactically incorrect (the grammar is wrong or the vocabulary is unknown). In order for the analyzer to be robust, it must then indicate to the user that LandScan is unable to answer the question because the query is ill-formed.

In order to produce output in some form useful to the reasoning processes it will be necessary for this module to have certain knowledge available. This includes the grammar of the language (English queries), the vocabulary necessary for this domain, and the semantics of locative constructs in order to produce a symbolic representation of the question.

5.4. Reasoning

The final component of LandScan is the reasoner which performs all high-level scene understanding operations. The reasoning operations are divided into three major categories (which are not nearly as simple as they appear):

1. find an object in the scene model
2. find an object part
3. find the spatial relations among objects

It is in the reasoner that all the parts of the system are tied together. In order to obtain the information necessary for the generation of the response the reaoner must have available to it both global knowledge and runtime data. The global knowledge includes the World Model and the Object Model (described below). The runtime data includes the sensory information available from the vision system in the form of the Scene Model (described above) and a symbolic representation of the query.

If the reasoning processes fail to produce a positive response (the query fails to have an answer), the reasoner performs two types of query failure analysis. The first type of query failure involves a query violating the global knowledge embodied in the World or Object Model. In this case, the system will respond with a message indicating that the query is conceptually ill-formed in this domain and why it is ill-formed. For instance, if the query asked how many walls the street had, the system would respond that streets do not have walls and that for that reason, the query is ill-formed. The other type of failure involves not finding the information requested in the scene model. In this case, rather than simply responding that the system was unable to find the data in question, it would prompt the vision system to move the cameras and combine this new view of the scene with the old one in order to obtain a positive response to the query.

5.4.1. World Model

Like the hierarchical relational model of Shapiro and Haralick, the world model describes the features and relations of the objects in the domain. The objects are those which can be expected in an urban scene - buildings, streets, sidewalks, etc. The world is represented by a labelled directed multigraph in which the nodes are the objects in the domain and the arcs are labelled with the relation which can hold between the two objects in the world. It has been determined that at least two relations are needed to adequately model the world. The two relationships are NEXT_TO and ON. NEXT_TO implies that two objects can be expected to be adjacent in the domain. This adjacency does not necessarily mean that the two objects will be adjacent in the geometric sense - sharing a common boundary - but that they would be viewed as being "close enough" to be considered adjacent. For example, CAR NEXT_TO BUILDING could be said to hold even if the car and building are separated by some other *small* object such as a sidewalk. The ON relation has one interpretation "on top of".

5.4.2. Object Model

The object model represents the expected physical features and linguistic properties of the objects in the domain. The physical properties are the parts of objects. These "parts" are the objects which were not included in the world model in order to keep the level of abstraction in that model consistent. In the object model, objects are decomposed into their possible parts.

The linguistic properties are those features which affect the usage and interpretation of a spatial construct (phrases describing the spatial relations between objects). Since the domain is a visual one, each object in the domain will have a "place" associated with it. This is what Herskovits calls the canonical geometric description of a spatial entity (objects) [Herskovits82].Ordinary solid objects (buildings, vehicles, people) are bounded closed surfaces. Geographical objects are entities with slightly imprecise boundaries - roads, rivers, and fields. Some other properties which must be represented are a prototype shape and the allowable deviations from it, the relative size, and characteristic orientation - ie. a table stands on its legs normally. The typical geometric conceptualization will also affect the choice of spatial construct - is the object normally viewed as a point or line. Along with the typical geometric conceptualization is the typical physical context of an object. For instance, a door is normally viewed as being in a wall. The normal function of an object, its functionally salient parts and the actions commonly performed with an object will also be necessary for analyzing the spatial constructs.

6. Future Work

In the data-driven system, much work still needs to be done in interpolating the depth map, edge fitting, and finding closed contours. In particular, it proves to be difficult to extract closed contours from the interpolated depth map (Section 4.8). Our future work will look hard at the feedback available from the failure to close contours and how it may be applied to the camera controller to take images to help close the contours. Other work concerns the implementation of algorithms for camera parameter control, corner detection, and measurement of focus sharpness.

In the query-driven system, the recognizer and scene model will provide the image information necessary to perform scene analysis of urban environments. The reasoning operations and linguistic analyzer must be fully specified and implemented. In particular, we will be paying special attention to the encoding of spatial prepositions and providing a computational model for handling query failure.

Before the scene model can be used in the query-driven image analysis process, the global knowledge embodied in the world and object models must be encoded. The world model will be represented graphically as explained above. The object model presents more of a challange though. The PART_OF relation has been handled in many systems [rosenthal81], [shapiro84]. However, no one has yet proposed a means of encoding the linguistic data which must be known about the objects in order to use them correctly in natural language utterances. Herskovits [herskovits82] suggests that certain object knowledge is relevant to the task of encoding and decoding locative constructions.

The natural language interface which uses the scene representation still has to be designed. It must be able to apply locative linguistic constructs to some representation of visual data and reason about this data. When this is operative, the scene analysis will be truly query-driven and the goals of the system will have been reached.

7. Conclusions

This paper has presented LandScan, a prototype vision system under development. This system covers most of the different levels of vision and natural language processing.

In summary, the data-driven subsystem of LandScan automatically acquires stereo images, enhances them by both linear and non-linear filtering, extracts edgels, matches edgels to generate a depth map, interpolates the depth map, fits edgels to depth points, uses the edges to build a surface grph, including geometric and topological attributes. The query-driven modules recognize objects and build a scene model which represents the user's interest in the image. It is controlled by the reasoner and linguistic analyzer which provide a computational model for handling spatial constructs and query failure.

While LandScan is not complete in the sense that all of it is successfully implemented, it provides a computational model for a vision system guided by natural language.

8. Acknowledgements

This research was supported by the following grants:

ARO DAA6-29-84-k-0061
AfOSR 82-NM-299
NSF MCS-8219196-CER
NSF MCS 82-07294
AVRO DAABO7-84-K-FO77
NIH 1-RO1-HL-29985-01

Also many thanks to Jon Korein for helping with SurfsUP.

9. Figures

Figure 1: Block diagram of LandScan.

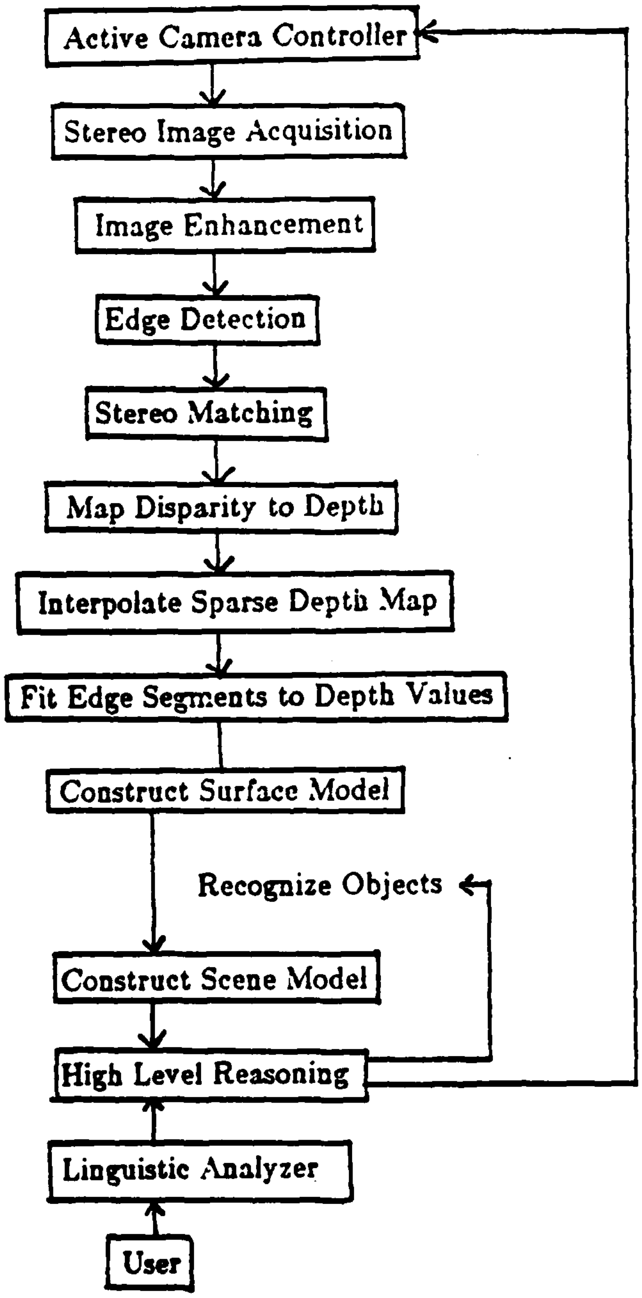


Figure 9-2: Stereo pair of aerial images of scale urban model, left and right. Objects include 3 buildings, 2 sidewalks, 1 tree, 1 field, and 2 roads.

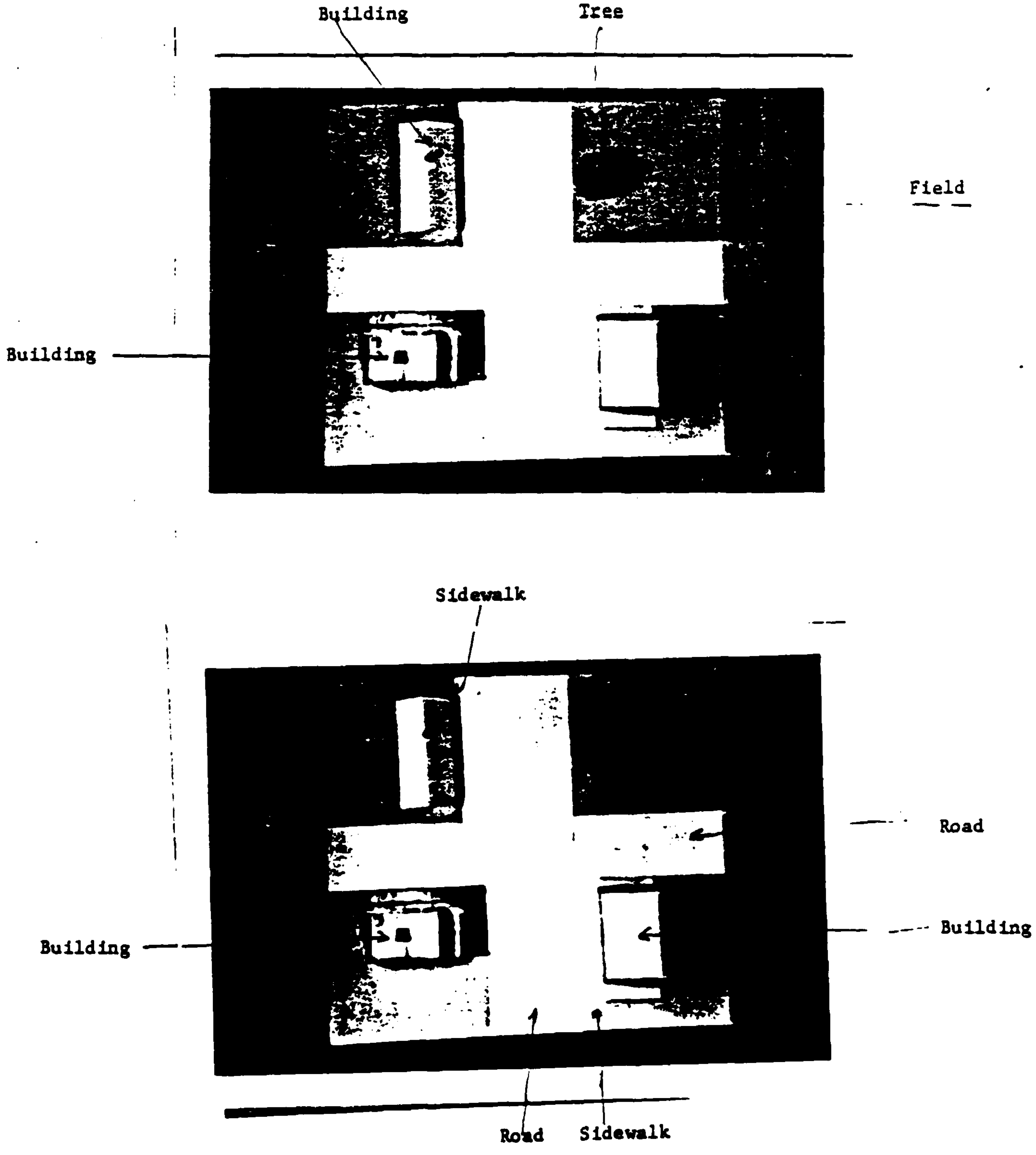


Figure 0-3: Results of Canny edge detector: left and right edgel maps.

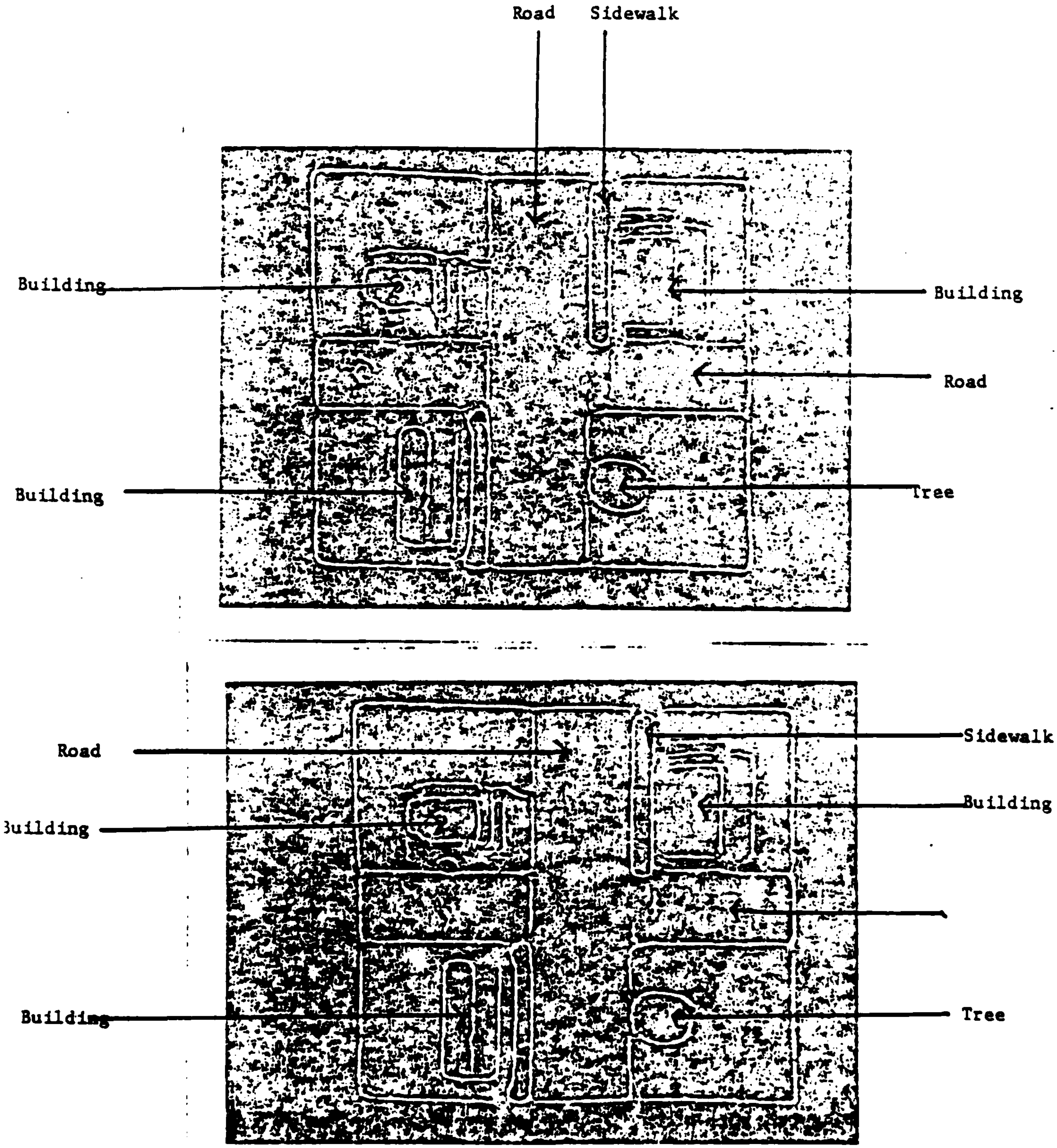


Figure 9-4: Edgels matched using 2-sided correlation.

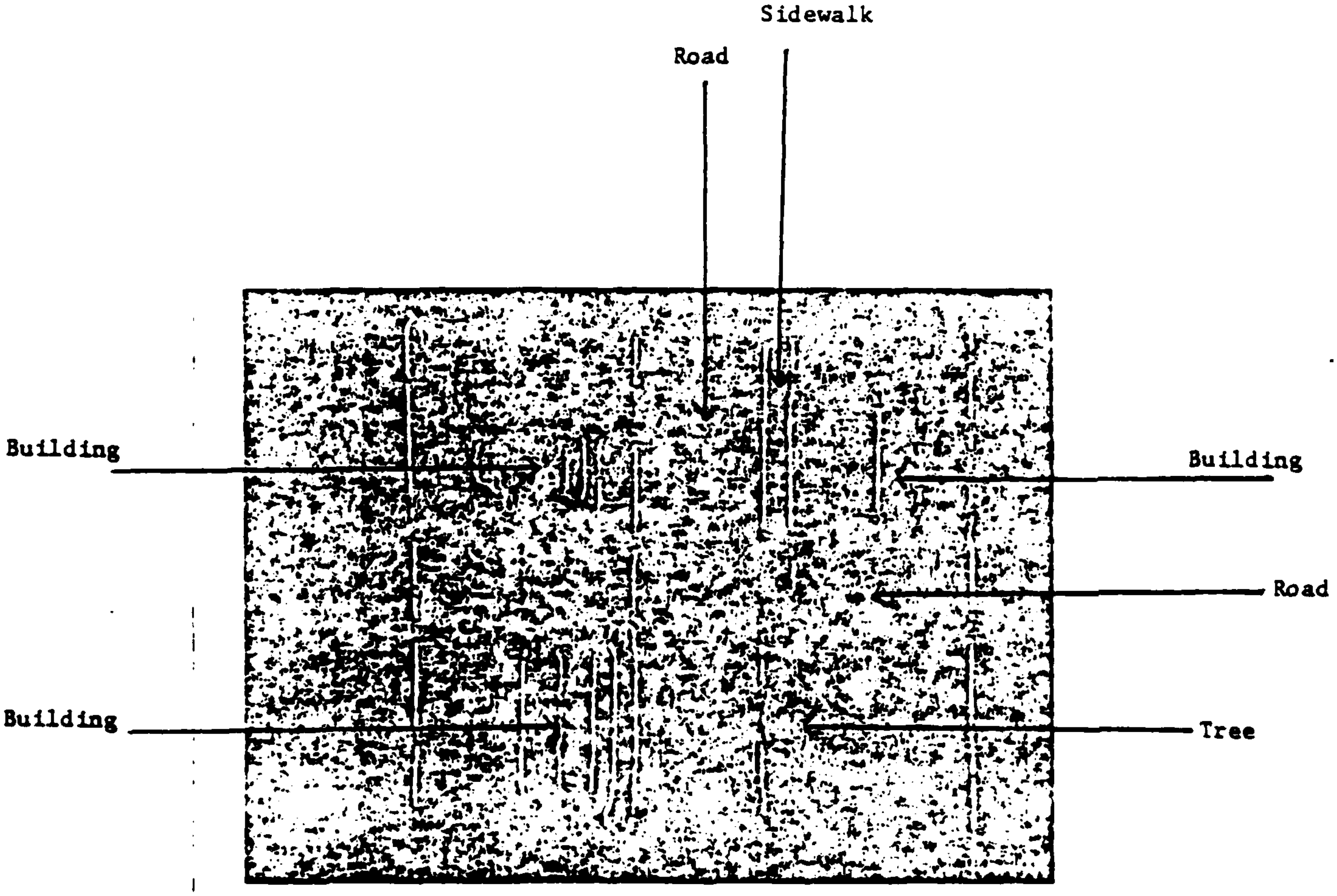


Figure 9-3: Edge segments generated from an interpolated depth map, corners specified interactively

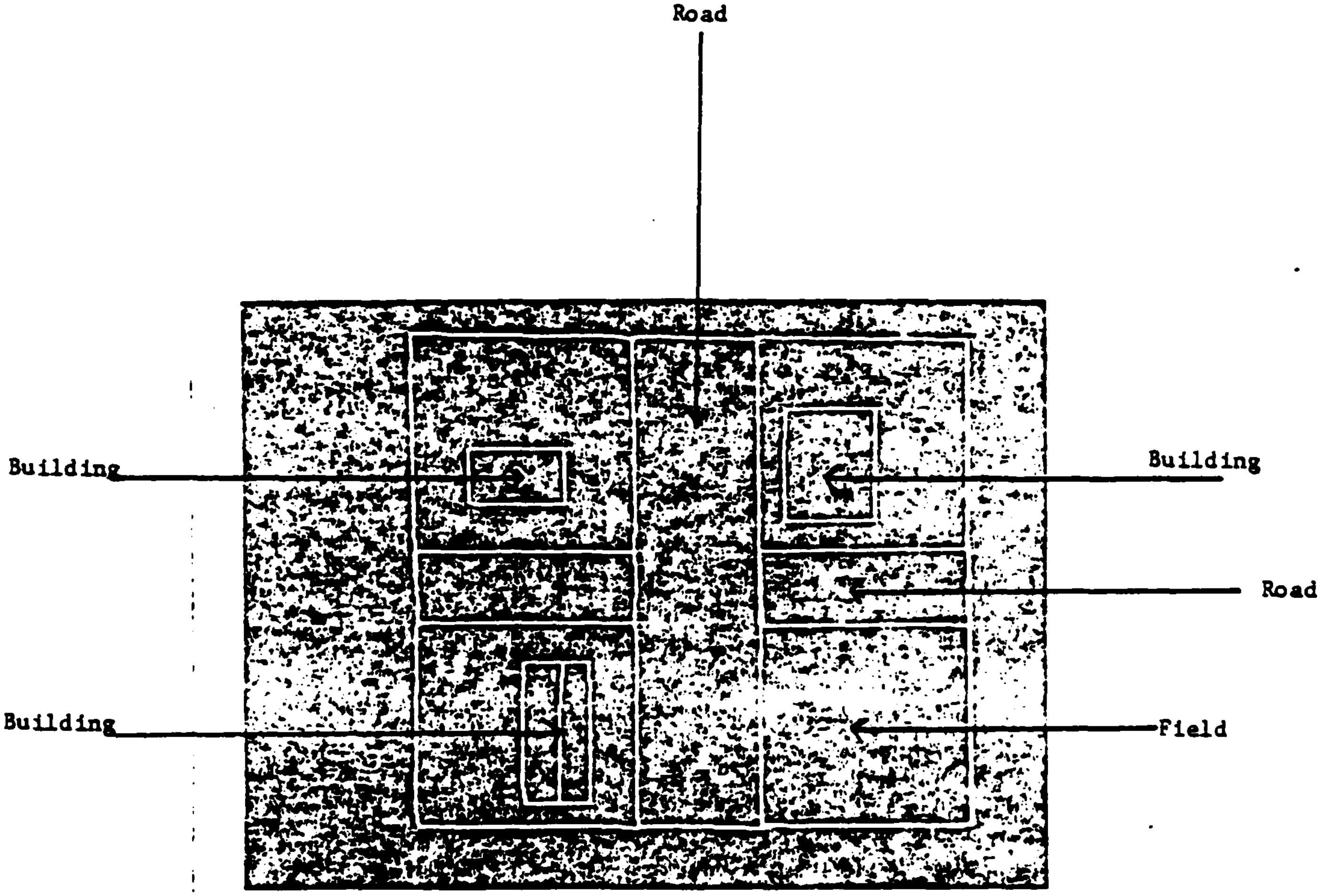


Figure 3-6: Reconstructed planar surfaces, rendered by Movie.BYU in top view.

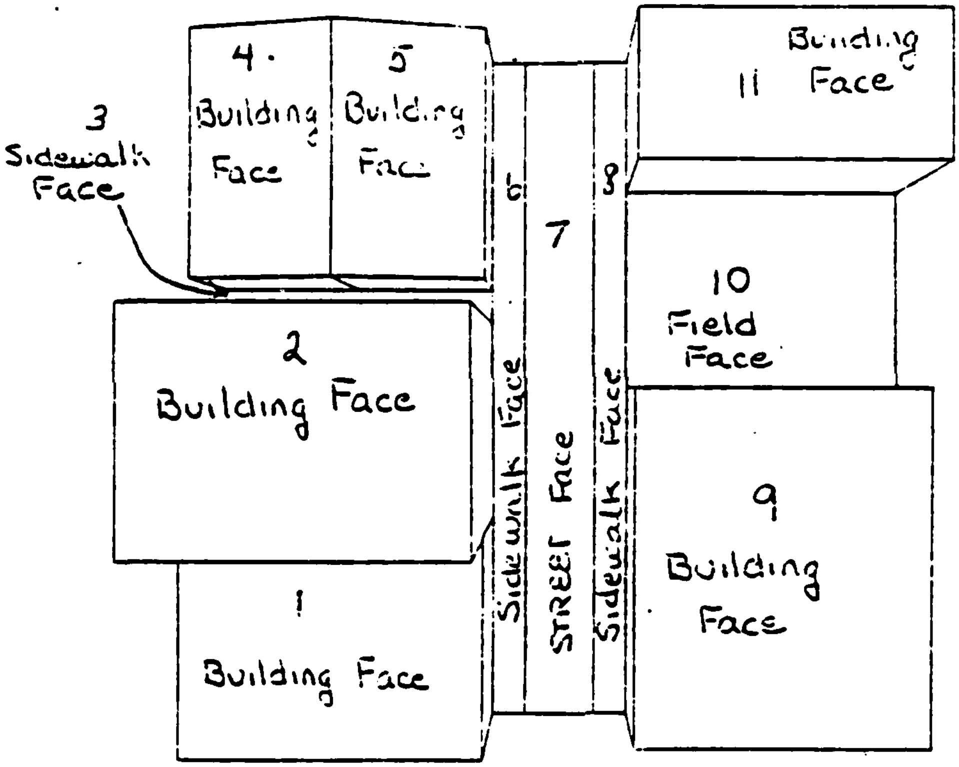


Figure 9-7: Objects recognized in FIGURE 6.

```
Lisp> (generate 'building)
object: BUILDING
subtype: COMPLEX
faces: 1 2

CONTIGUOUS
0

ADJACENT
0

LOOKSCONTIGUOUS
0

LOOKSADJACENT
0

ABOVE
0

CONTAINS
0

Lisp> (generate 'sidewalk)
object: SIDEWALK
subtype: COMPLEX
faces: 3 6

CONTIGUOUS
0 0
0 0

ADJACENT
0 0
0 0

LOOKSCONTIGUOUS
0 1
1 0

LOOKSADJACENT
0 1
1 0

ABOVE
0 1
0 0

CONTAINS
0 0
0 0
```

```
Lisp> (generate 'building)
object: BUILDING
subtype: HOUSE
faces: 4 5

CONTIGUOUS
0 0 0
0 0 0
0 0 0

ADJACENT
0 0 0
0 0 0
0 0 0

LOOKSCONTIGUOUS
0 1 0
1 0 1
0 1 0

LOOKSADJACENT
0 1 0
1 0 1
0 1 0

ABOVE
0 1 1
0 0 0
0 1 0

CONTAINS
0 0 0
0 0 0
0 0 0

Lisp> (generate 'street)
object: STREET
subtype: SIMPLE
faces: 7

CONTIGUOUS
0 0 0 0
0 0 0 1
0 0 0 0
0 1 0 0

ADJACENT
0 0 0 0
0 0 0 1
0 0 0 0
0 1 0 0
```

```
LOOKSCONTIGUOUS
0 1 0 0
1 0 1 1
0 1 0 0
0 1 0 0

LOOKSADJACENT
0 1 0 0
1 0 1 1
0 1 0 0
0 1 0 0

ABOVE
0 1 1 1
0 0 0 0
0 1 0 1
0 0 0 0

CONTIGUOUS
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

References

[Akey 84] M. L. Akey , O. R. Mitchell. Detection and Sub-Pixel Location of Objects in Digitized Aerial Imagery. In *Seventh International Conference on Pattern Recognition*, pages 411-414. July 30-August 2, 1984.

[Bates 81] Bates, Madeleine. The Theoty and Practice of Augmented Transition Network Grammars. In Leonard Bolc (editor), *Natural Language Communication with Computers*. Springer-Verlag, 1981.

[Canny 84] John F. Canny. *Finding Edges and Lines in Images*. Technical Report AI-TR-720, MIT, 1984.

[Faugeras 81] Olivier D. Faugeras . Keith Price. Semantic Description of Aerial Images Using Stochastic Labeling. *IEEE Trans. on PAMI* PAMI-3(4):459-469, July, 1981.

[Fischler 83] Martin Fischler. *Image Understanding Research and its Application to Cartography and Computer-Based Analysis of Aerial Imagery*. Technical Report, SRI International, September, 1983.

[Glicksman 83] Glicksman, Jay. Using Multiple Information Sources in a Computational Vision System. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. 1983.

[Grimson 81] W.E.L. Grimson. *From Images to Surface: A Computational Study of the Human Early Vision System*. MIT Press, 1981.

[Harlow 84] C. A. Harlow , R. W. Conners , M. Trivedi. A Computer Vision System for the Analysis of Aerial Scenes. In *Seventh International Conference on Pattern Recognition*, pages 407-410. July 30-August 2, 1984.

[Herman 83] Herman, Martin, Takeo Kanade, Shigeru Kuroe. The3D MOSAIC Scene Understanding System. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. 1983.

[Herskovits 82] Herskovits, Annette. *Space and the Prepositions in English: Regularities and Irregularities in a Complex Domain*. PhD thesis, Department of Linguistics, Stanford University, 1982.

[Herskovits 84] Herskovits, Annette. Space and the Prepositions in English: Regularities and Irregularities in a Complex Domain. 1984.Draft: University of California, Berkeley.

[Hwang 83] Hwang, Vincent, Takashi Matsuyama, Larry Davis, Azriel Rosenfeld. *Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding*. Technical Report, Center for Automation Research, University of Maryland, October, 1983.

[Izaguirre 84] Alberto Izaguirre , Pearl Pu , John Summers. *A New Development in Camera Calibration -- Calibrating a Pair of Mobile Cameras*. Technical Report MS-CIS-84-55, University of Pennsylvania, 1984.

[Krotkov 84] Krotkov, Eric. *Construction of a Three Dimensional Surface Model*. Technical Report, GRASP LAB, CIS Department, University of Pennsylvania, 1984.

[Lee 85] Yong Hoon Lee, Saleem A. Kassam. Generalized Median Filtering and Related Nonlinear Filtering Techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33, accepted for publication in 1985.

[Marr 80] David Marr, Ellen Hildreth. Theory of Edge Detection. In *Proc. R. Soc. Lond.*, pages 187-217. 1980.

[Nagao 79] M. Nagao , T. Matsuyama , H. Mori. Structural Analysis of Complex Aerial Photographs. In *IJCAI 6*, pages 610-617. August 20-23, 1979.

[Nevatia 83] Ramakant Nevatia. *Final Technical Report.* Technical Report Report 104, Intelligent Systems Group, University of Southern California, October 19, 1983.

[Potmesil 83] Potmesil, Michael. Generating Models of Solid Objects by Matching 3D Surface Segments. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[Quam 78] L. H. Quam . *Road Tracking and Anomaly Detection in Aerial Imagery.* Technical Report Technical Note 158, SRI International, March 1978.

[Radack, et al 84]
Radack, Korein, Ganis, McNally, Korein, Shapiro. *NASA Programmer's Guide* CIS Department, University of Pennsylvania, 1984.

[Reynolds, et al 84]
Reynolds, G, N.Irwin, A.Hanson, E.Riseman. Hierarchical Knowledge-Directed Object Extranction Using a Combined Region and Line Representation. In *Vision Workshop.* 1984.

[Rosenthal 81] Rosenthal, David. *An Inquiry Driven Vision System Based on Visual and Conceptual Hierarchies.* UMI Research Press, 1981.

[Shapiro 84] Shapiro and Haralick. A Heirarchical Relational Model for Automated Inspection Tasks. In *Int. Conf. on Robotics, Atlanta, Ga..* 1984.

[Sloan 81] Kenneth R. Sloan , P. G. Selfridge. Reasoning about images: Applications to aerial image understanding. In *Proc. 1981 Image Understanding Workshop*, pages 1-6. April 1981.

[Smitley 84] David L. Smitley , Ruzena Bajcsy. Stereo Processing Aerial, Urban Images. In *Seventh International Conference on Pattern Recognition*, pages 433-436. July 30-August 2, 1984.

[Talmy 83] Talmy, Leonard. *How Language Structures Space.* Technical Report 4, Berkeley Cognitive Science Report, January, 1983.

[Talton 84] David Talton. *Implementation of a Gaussian-Smoothing Gradient-Based Edge Detector.* Technical Report MS-CIS-84, University of Pennsylvania, 1984.

[Tropf 83] Tropf and Walters. An ATN for 3-D Recognition of Solids in Single Images. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* 1983.

[Winograd 83] Winograd, Terry. *Language as a Cognitive Process.* Addison-Wesley Publishing Co., 1983.

[Winston 79] Winston, Patrick Henry. *Artificial Intelligence.* Addison-Wesley Publishing Company, 1979.

APPENDIX 1
LandScan: A Computer Vision System for Analyzing Aerial Images

Eric Krotkov
Ruzena Bajcsy

CIS Dept/D2
University of Pennsylvania
Philadelphia, Pennsylvania 19104
(215) 898 - 6222

Abstract

This paper presents LandScan (LANguage Driven SCene ANalysis) as an integrated vision system which covers many levels of both vision and natural language processing. Computations are both data-driven and query-driven, but only the data-driven system is considered here.

The data-driven system employs active control of stereo cameras for image acquisition, and a bottom-up flow of control dynamically constructs a surface model from multiple aerial views of an urban scene. Processing steps include: image acquisition, image enhancement, edge detection, stereo matching, mapping disparity to depth, interpolation of sparse depth points, edge segment fitting, construction of surface model, including surface attributes and topological relations between surfaces.

LandScan:
A Computer Vision System for Analyzing Aerial Images

1. Introduction

The aim of our research is to develop a system capable of dynamically updating and maintaining a model of an urban world over multiple aerial views. The system will have a natural language front end [Zwarico 84], through which users can query the system about what it sees, and to direct or interactively assist the vision processing by restricting the analysis to those areas of the scene which are of current interest. The representation is dynamic, constructed as the system is queried, and explicitly represents the history of the user's interest in the scene.

A unique contribution of the work is that processing is both data-drivenn ("bottom up," determined by sensor data) and query-driven ("top down," determined by user queries). The integration of both methods into one system can help overcome the shortcomings of each method employed independently. For example, if data-driven processing were able to segment a graph of edges derived from the image into *several* different connected components, query-driven information about what the system *should* be looking for can help impose structure, and a unique segmentation, upon the otherwise ambiguous data.

The data-driven processing starts with stereo aerial images and proceeds, by filtering, matching, interpolating, and fitting, to reconstruct the surfaces in the scene. The aerial domain buys the simplicity of *planar* surfaces. Two factors distinguish this data-driven system from many others. First, image acquisition is controlled by feedback from the query-driven system, and is undertaken by *active sensors*, actively probing the environment. Second, the controlled environment of a scale urban model allows precise verification of results and proper operation of individual modules.

This paper will describe the aerial domain, some related research, the implementation of the data-driven portion of the LandScan (LANguage Driven SCene ANalysis) system, and our plans for future work.

2. The Aerial Domain

Aerial images suffer from a poverty of context, due to the distance at which images are formed. Urban scenes contain featureless areas and large numbers of occlusion edges. Even with the best possible use of image data, we generally can do no better than to compute a sparse depth map of the imaged scene. For many purposes a sparse depth map is inadequate, and the missing surface information must be obtained from other sources: other "shape from ..." processes, domain-dependent high-level knowledge, and real-world constraints.

There are two major constraints in the aerial domain:

1. The data is obtained by taking *aerial* photographs of an *urban* environment. Urban scenes are characterized by an abundance of straight lines. This means that to a very good approximation the scene, as viewed from on high, is composed of planar polyhedra, so that detected edges separate planar surfaces, i.e., each edge arises because it is the intersection of 2 planar faces.

2. The image acquisition process is under our control, so the camera model is known. Some combination of azimuth and elevation angles, Euler angles, pan, roll, tilt angles are available and fully specify a 4x4 homogeneous transformation relating the position and orientation of the two cameras.

Domain knowledge includes such facts as roofs of buildings tend to be parallel to the ground plane, while walls are perpendicular to it, and that sidewalks are thinner (more compact) than roads.

3. Related Research

A large corpus of research on aerial image understanding *per se* exists, [Akey 84], [Harlow 84], [Hwang 83], [Nagao 79], [Sloan 81], [Quam 78], [Faugeras 81], and many general vision techniques are applicable to the aerial domain. Large aerial projects have been undertaken at USC [Nevatia 83] and at SRI [Fischler 83].

The 3D MOSAIC project [Herman 83] is geared toward the urban aerial domain. Important differences in their strategy are that junctions are primitive, and a monocular analysis is performed. At the level of object representation LandScan treats surfaces as primitive, while 3D Mosaic treats faces, edges and points as primitives. Further, in 3D MOSAIC hypotheses are generated about the continuation of occluded lines, shapes of faces, and the extent of vertical faces. The construction of the scene model in the 3D MOSIAC system is exclusively data-driven, while LandScan uses a query-driven approach for constructing the Scene Model.

Shapiro and Haralick [Shapiro 84] describe a hierarchical, relational 3D model which is influential in our design. Their model provides precise, accurate information to be used by low-level vision and inspection processes as well as information required by high-level vision and reasoning processes. All of the information is represented by using "spatial data structures", each consisting of a recursive set of relations. The hierarchy consists of four levels: world, object, part, and surface/arc.

4. System Implementation and Results

This section will describe the implementation and performance of the major data-driven modules, illustrated in Figure 8-1: image acquisition, image enhancement, edge detection, stereo matching, mapping disparity to depth, interpolation of sparse depth points, edge segment fitting, construction of surface model, including surface attributes and topological relations between surfaces.

4.1. Image Acquisition

Presently images are acquired manually by positioning cameras above a scale model of some toy buildings. Figure 8-2 illustrates a typical stereo pair of images acquired. A system for automatically setting camera parameters (location, pan, tilt, focus, zoom, aperture, vergence angle) has been constructed, and a controller for optimizing the parameters on the basis of feedback from high-level goals, medium-level strategies, and low-level image features is under design. This "smart camera" is an active sensor, capable of moving in or out for a better look, zooming in on a feature, improving its signal/noise ratio, and much more. Our philosophy is to have the sensors do as much of their own processing as possible in a heterarchical environment.

4.2. Image Enhancement

Before extracting the features for matching the images are smoothed with a non-linear double window median filter [Lee 85], removing impulse noise, and suppressing high-frequency noise. Independently, the range of grey scales is extended to cover 256 values by linear contrast enhancement (see Figures 8-3 and 8-4).

4.3. Edge Detection

Two edge operators have been implemented, the Canny [Canny 84], [Talton 84] and DOG zero-crossing [Marr 80] methods. As implemented, these operators return "edgels," defined as points possibly lying at an intensity discontinuity, rather than "edges," defined as a set of edgels lying along a space curve. The two operators are now being carefully evaluated and compared on the basis of false positives, false negatives, and overall robustness under focus degradation and illumination degradation. Although the verdict is not yet in, the Canny operator is presently employed, and typical results are shown in Figure 8-5.

The Canny operator approximates a directional first derivative. The direction information can be used to find areas of high curvature (e.g., corners). Our present approach is to look at the variance of the directions in a small edgel neighborhood to identify corners and junctions.

4.4. Stereo Matching

Because of the large interocular distance in the aerial (fly-by) imaging there are large disparity jumps and large portions of the scene are visible in one image but not the other. This occlusion problem has haunted many matchers.

The matcher [Smitley 84] employs the method of 2-sided correlation in order to circumvent some of the difficult problems of occlusion, and uses a registration technique to bring the scan lines into correspondence [Izaguirre 84]. Figure 8-6 illustrates the results from the matcher. Present work in matching concerns evaluating its robustness, extension to higher-order features (e.g., linear segments, corners, and junctions), and obtaining horizontal disparities as well by taking three views per stereo frame instead of two.

4.5. From Disparity to Depth

Generally both disparity (distance in image space between matching pixels) and depth (distance in 3-space from viewer to object) are measured in a viewer-centered coordinate system. The function from disparity to depth (absolute, not relative) is linear in the disparity, interocular distance, focal length, and vergence angle. In the case where the view vector is parallel to the ground, a large disparity implies that the object is close, i.e., has a *small* depth value. In the case where the view vector is perpendicular to the ground (i.e., in the aerial domain) a large disparity implies that the object is close, i.e., is far from the ground. We adopt the convention of mapping large disparities into large depths.

The method is essentially triangulation. We are building hardware to both control and measure the vergence angle between two cameras. With this angle, the exact distance to any point fixated in both visual fields can be discovered. Given this exact distance, the relative depth map returned from stereo can now be fixed as an absolute depth map.

4.6. Depth Point Interpolation—Filling In The Gaps

The research issue for any scheme of filling the gaps is the trade-off between the measurements and the *a priori* information. We elaborate this trade-off with an example. Let us suppose that we have a sparse array of 3D points after a stereo and/or optical flow computation. Remember we are left with some points that have not been matched either in the stereo matching nor in optical flow computation. In order to fill in the gaps we have several possibilities:

a) we can ignore the unmatched points, i.e., have confidence only in those points (measurements) that have been matched. Then assume, let us say a linear (or any polynomial) model (the *a priori* information about the local surface). Based on this we perform linear (or polynomial) interpolation between the neighboring points.

b) an alternative to the case (a) is instead of assuming the linear or polynomial model, which are inherently local, neighborhood models, assume a global smoothness constraint, which using variational calculus tries to fit the smallest and smoothest surface over the sparse data [Grimson 81].

c) the third possibility is to assume a local smoothness constraint in the depth values. Then reexamine the unmatched points (match them with the closest edgels in the other image) and check whether their depth value would satisfy the smoothness constraint with the neighboring points.

d) Finally if, for example, from the outline we can identify measured object then clearly the "fill in gaps" process can use this information. Example of this case can be sidewalks or roads in aerial views.

As usual in machine perception, there is no one technique that works uniformly well in all cases. We believe that this is an integral part of the surface interpretation. One clearly needs all the above techniques available and then having a rule-based system use whichever give the "best" results. For example if we have one obejct in the view, then perhaps the third method is the "best". If one has reason to assume that one deals with objects that have only planar surfaces, then the first method might be adequate. The third method is the most versatile since it uses the most measurements and the least *a priori* information. The cost is in computation.

Presently two types of interpolation are implemented. The first attempts to restore edgels which should have been matched, but were not matched, by comparing the depth map with a map of edgels with a largely vertical (hence matchable) component in its directional derivative. The depth map is updated by adding selected edgels with a linearly interpolated depth value (Figure 8-7). This is an important process, and the results of linear intepolation are not entirely satisfactory. Improved interpolation will use corners and junctions in the near future.

The second interpolation fills depth values in featureless areas. This is quite simple-minded [Grimson 81] and does not provide exceptional results (see Figure 8-8). But because this is used primarily for display purposes, i.e., we do not need to hypothesize about featureless areas because of the aerial perspective on an urban world, this is not a significant problem.

4.7. Edge Segment Fitting--Generating Wire Frames

This process fits a set of (straight) line segments in 2-point form (wire frames) from a rich set of depth points by a divide-and-conquer method of recursive decomposition. This method assumes that the boundary is of low curvature, and needs information about the location of corners to operate correctly. Figure 8-9 illustrates the edge segments generated from an interpolated depth map, and corners specified interactively.

4.8. Surface Model

A graph is constructed to serve as the surface model. This process converts a set of contours into a set of *closed* contours represented as a graph (a linked list of vertices, edges, and faces, as in Figure 8-10). The construction algorithm looks for minimum distance paths from a vertex back to itself, by traversing edges and at trihedral junctions choosing the path making the most acute angle with respect to the present path. Figures 8-11 and 8-12 illustrate the faces represented in the surface model.

Surface attributes and relations are computed in the SursUP [Radack, et al 84] geometrical modeling system. In it, a *face* is defined by its enclosing 3D contours. Attribute values for each face in the surface graph are computed [Krotkov 84]:

compactness, centroid vector, (outward-pointing) normal vector, area, "type," (building, sidewalk, field, street, and unknown), and number of sides. These values are computed once and stored on an attribute list.

Computed topological relations are *above*, *adjacent* (touching), *contiguous* (sharing an edge), *contains* (proper inclusion), *looksadjacent*, *lookscontiguous* (respectively adjacent and contiguous under perspective transformations) [Krotkov 84]. Relations (and indirectly their complements) are computed once and stored as Boolean arrays. These relations are expensive to compute because they require intersection operations (except the *above* relation).

5. Future Work

In the data-driven system, much work still needs to be done in interpolating the depth map, edge fitting, and finding closed contours. In particular, it proves to be difficult to extract closed contours from the interpolated depth map (Section 4.8). Our future work will look hard at the feedback available from the failure to close contours and how it may be applied to the camera controller to take images to help close the contours. Other work concerns the implementation of algorithms for camera parameter control, corner detection, measurement of focus sharpness, and using feedback from failure to recognize an object to guide future processing.

6. Conclusions

This paper has presented LandScan, an integrated vision system under development. This system covers most of the different levels of vision and natural language processing, integrating sensor information with surface, scene and world models.

In summary, the data-driven subsystem of LandScan automatically acquires stereo images, enhances them by both linear and non-linear filtering, extracts edgels, matches edgels to generate a depth map, interpolates the depth map, fits edgels to depth points, uses the edges to build a surface graph, including geometric and topological attributes.

While LandScan is not complete in the sense that all of it is successfully implemented, the system covers a wide spectrum of vision and natural language processing.

7. Acknowledgements

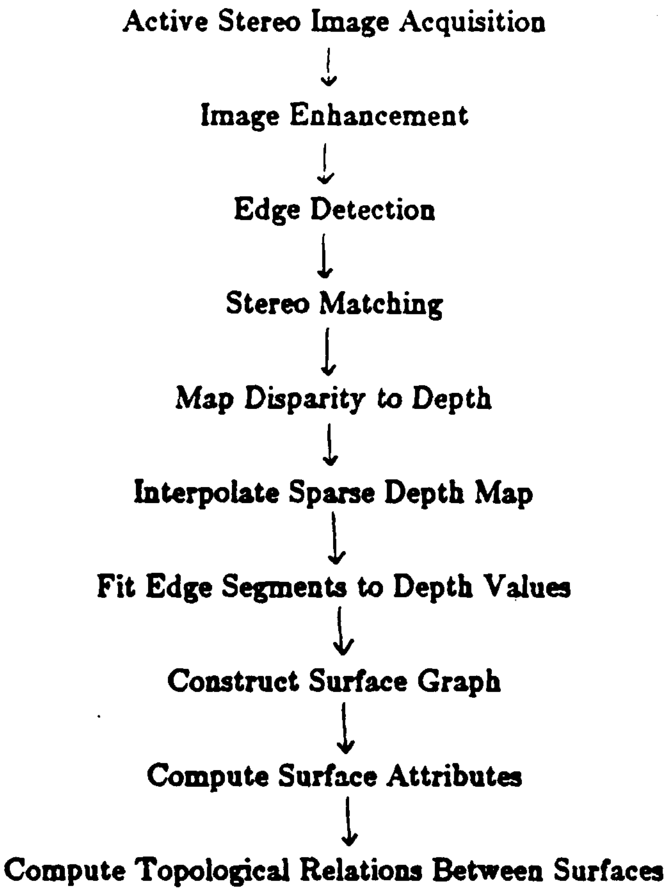
This research was supported by the following grants:

ARO DAA6-29-84-k-0061
AfOSR 82-NM-299
NSF MCS-8219196-CER
NSF MCS 82-07294
AVRO DAABO7-84-K-FO77
NIH 1-RO1-HL-29985-01

Also many thanks to Jon Korein for helping with SurfsUP.

8. Figures

Figure 8-1: Block diagram of data-driven section of LandScan system.



Active Stereo Image Acquisition

Image Enhancement

Edge Detection

Stereo Matching

Map Disparity to Depth

Interpolate Sparse Depth Map

Fit Edge Segments to Depth Values

Construct Surface Graph

Compute Surface Attributes

Compute Topological Relations Between Surfaces

Figure 8-2: Stereo pair of aerial images, left and right, of scale urban model. Objects present include 3 buildings, 2 sidewalks, 1 tree, 1 field, and 2 roads.

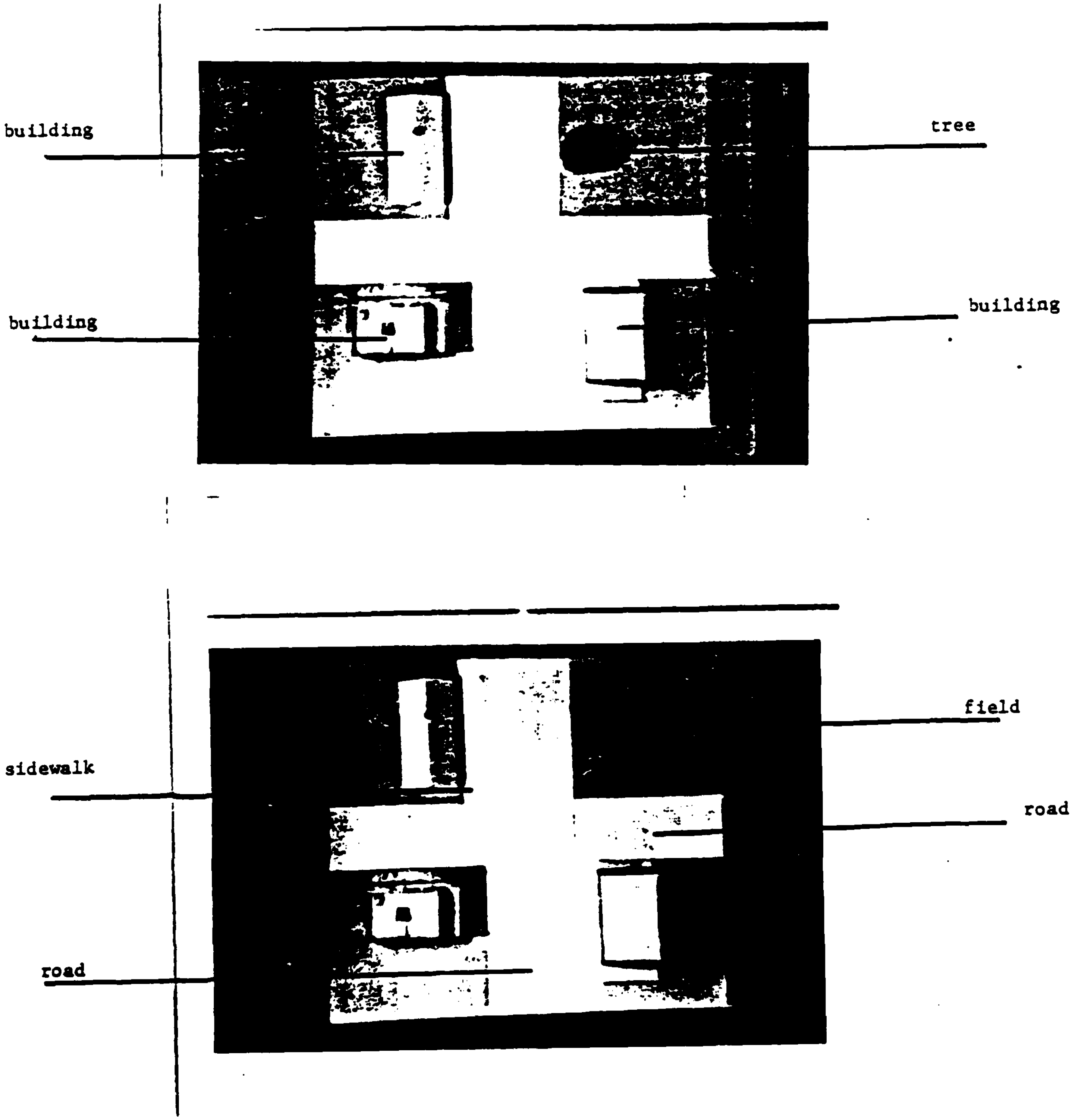


Figure 8-3: Histograms of left image intensities, before and after enhancement. The enhancement has caused some aliasing, but has improved the contrast by a factor of 1.5. Right image histograms are very similar.

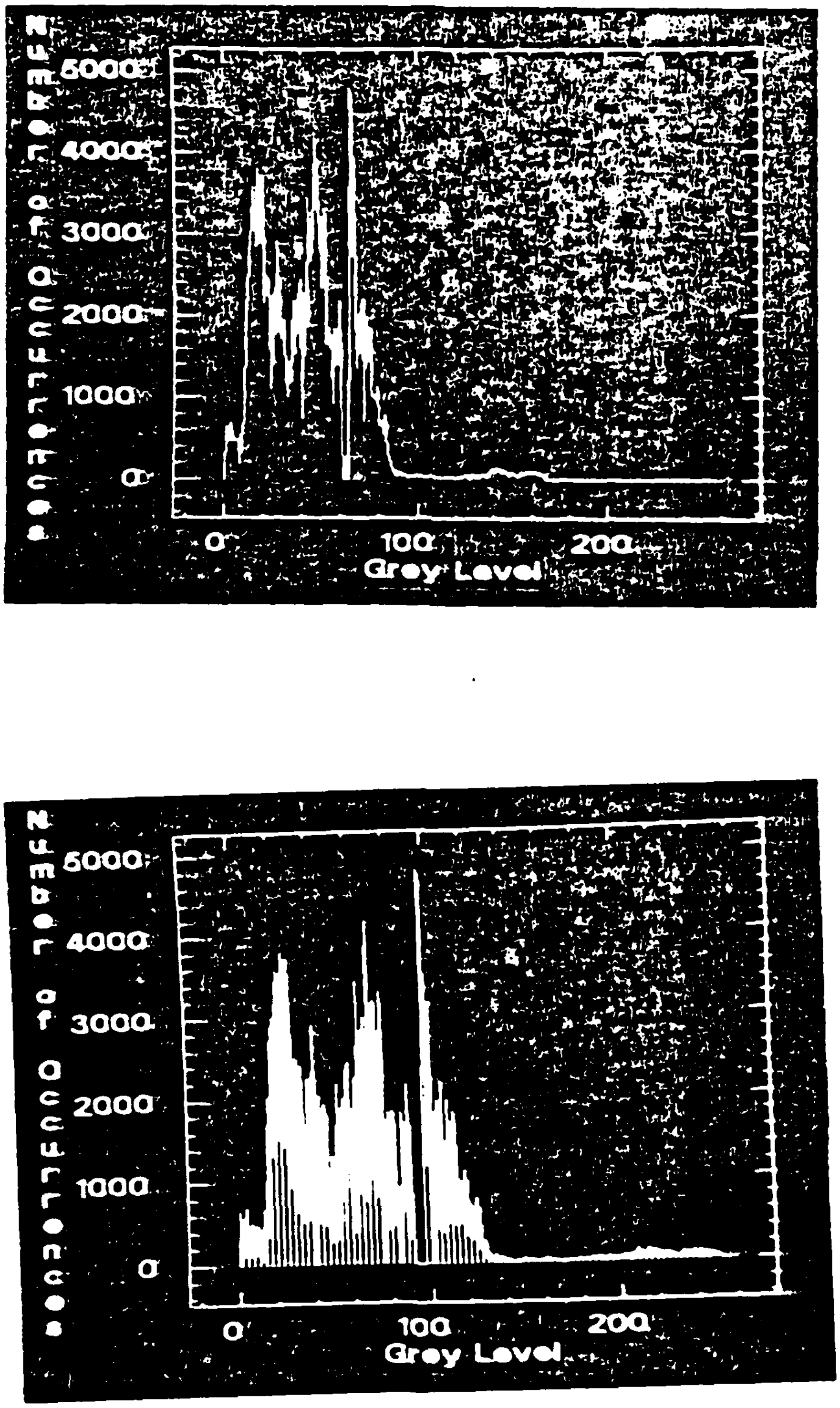


Figure 8-4: Enhanced images, left and right.

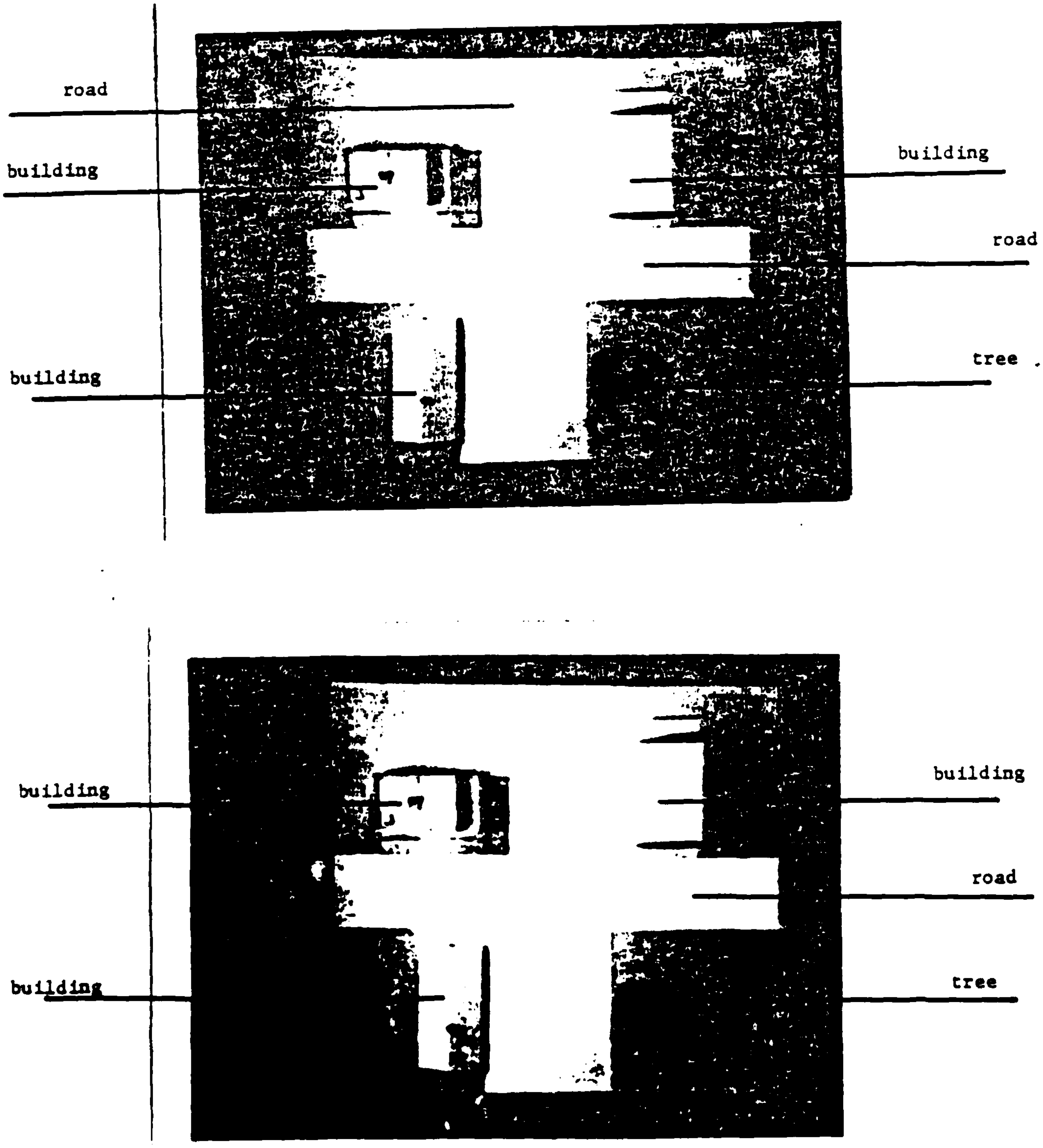


Figure 8-5: Results of Canny edge detector, left and right "edgel" maps.

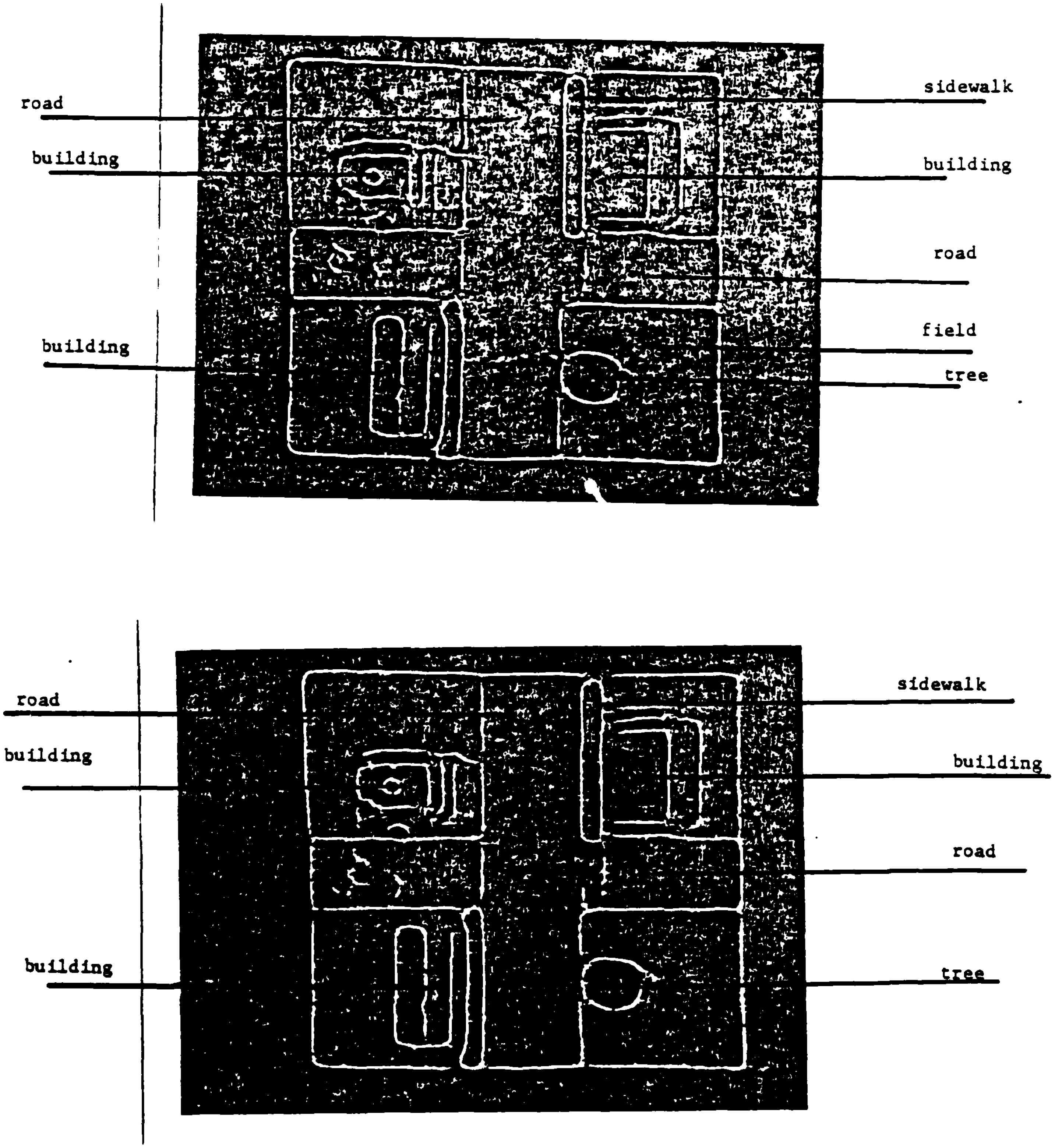


Figure 8-6: Edgels matched using 2-sided correlation.

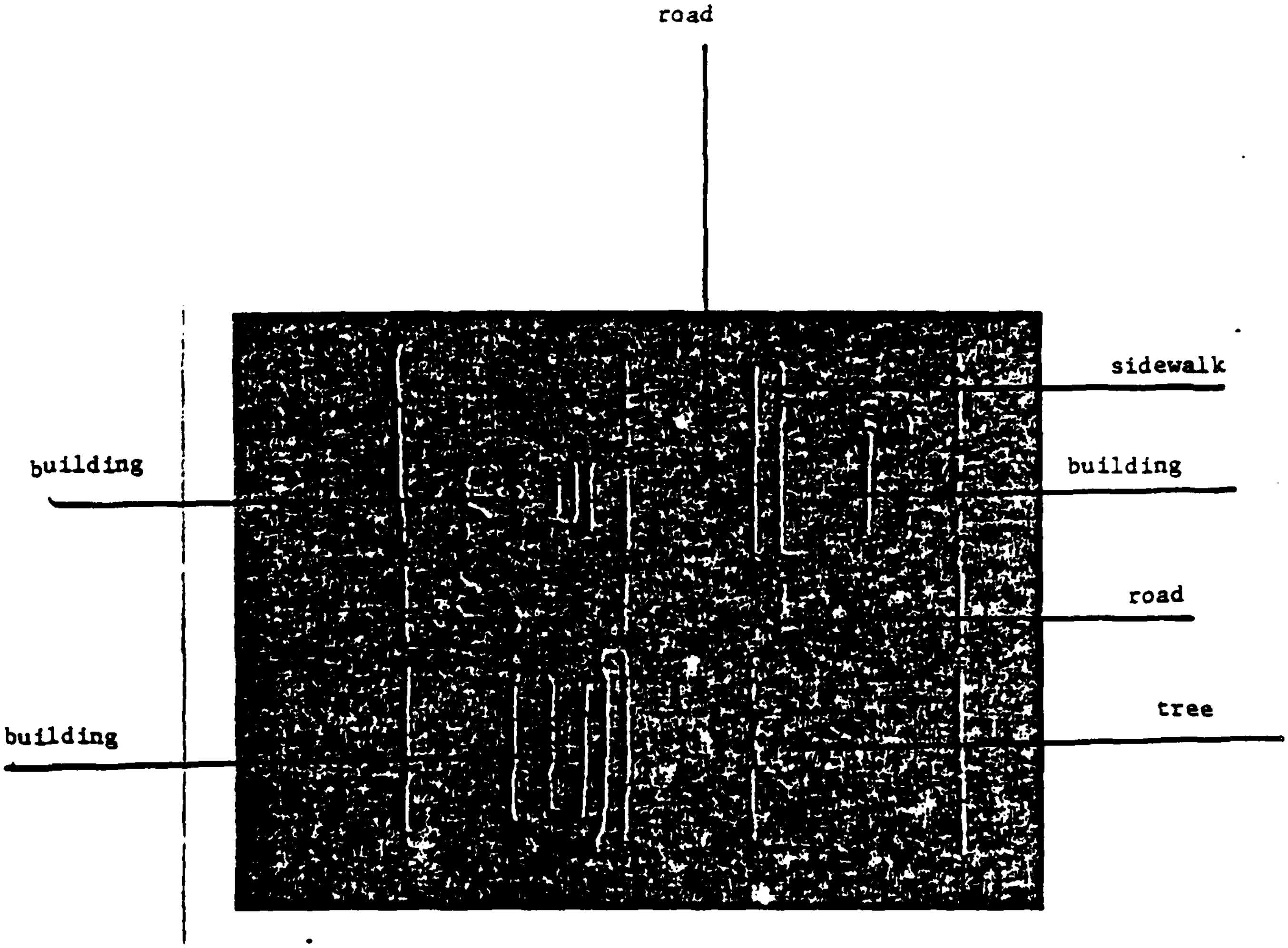


Figure 8-7: Depth map with depth values linearly interpolated. This binary picture depicts only the location of non-zero depths. Picture generated directly from disparities and original images.

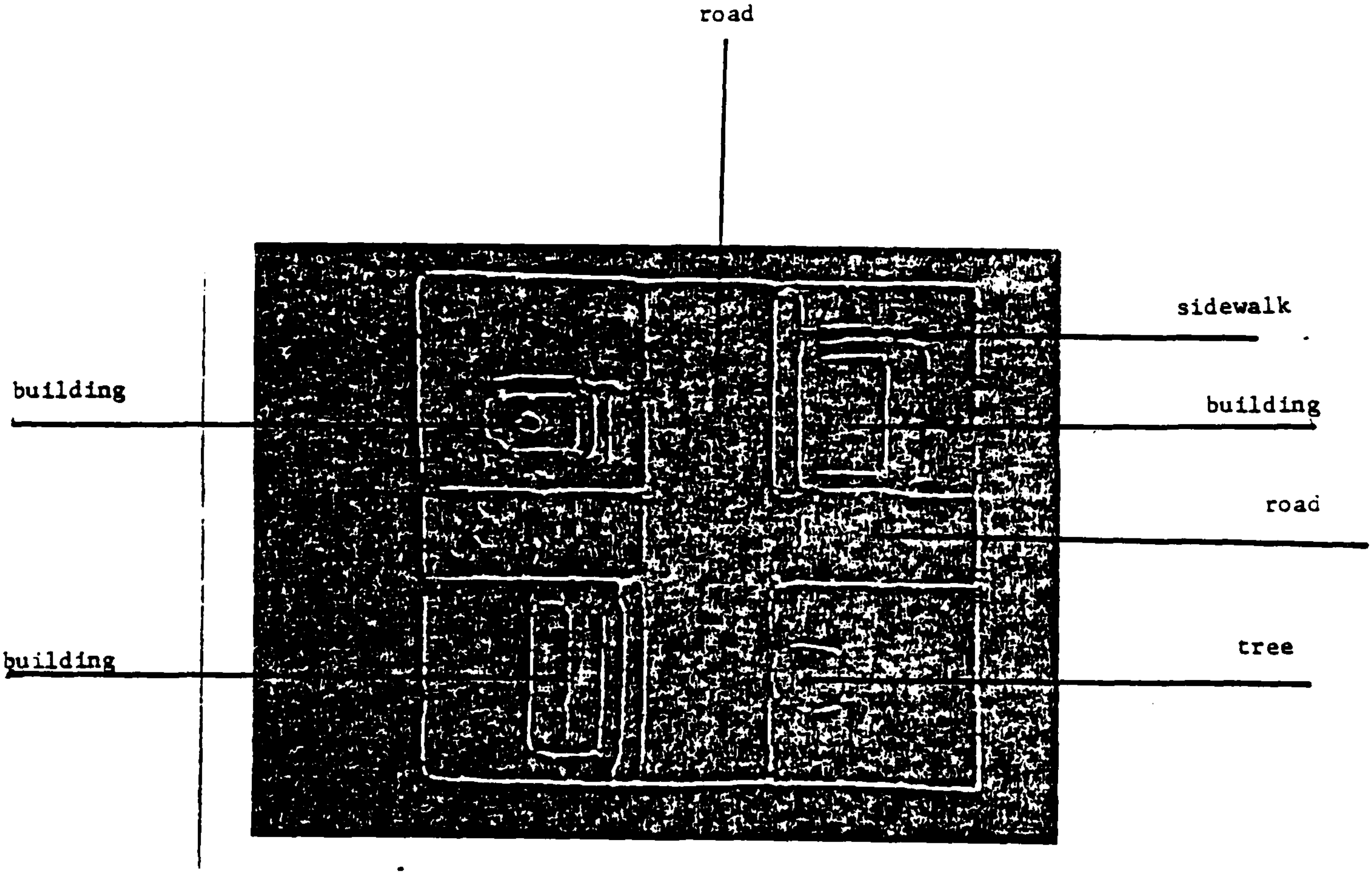


Figure 8-8: Depth map with depth values linearly interpolated inside of featureless areas. Picture generated directly from disparities and edgels. The long troughs are roads and sidewalks. The spikes are buildings.

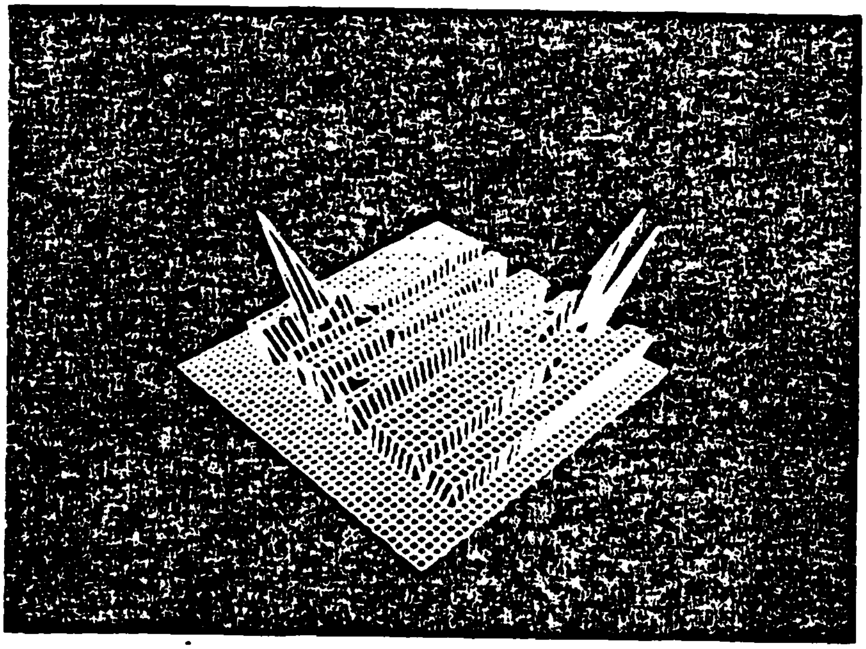


Figure 8-9: Edge segments generated from an interpolated depth map. This picture not generated automatically: corners specified interactively.

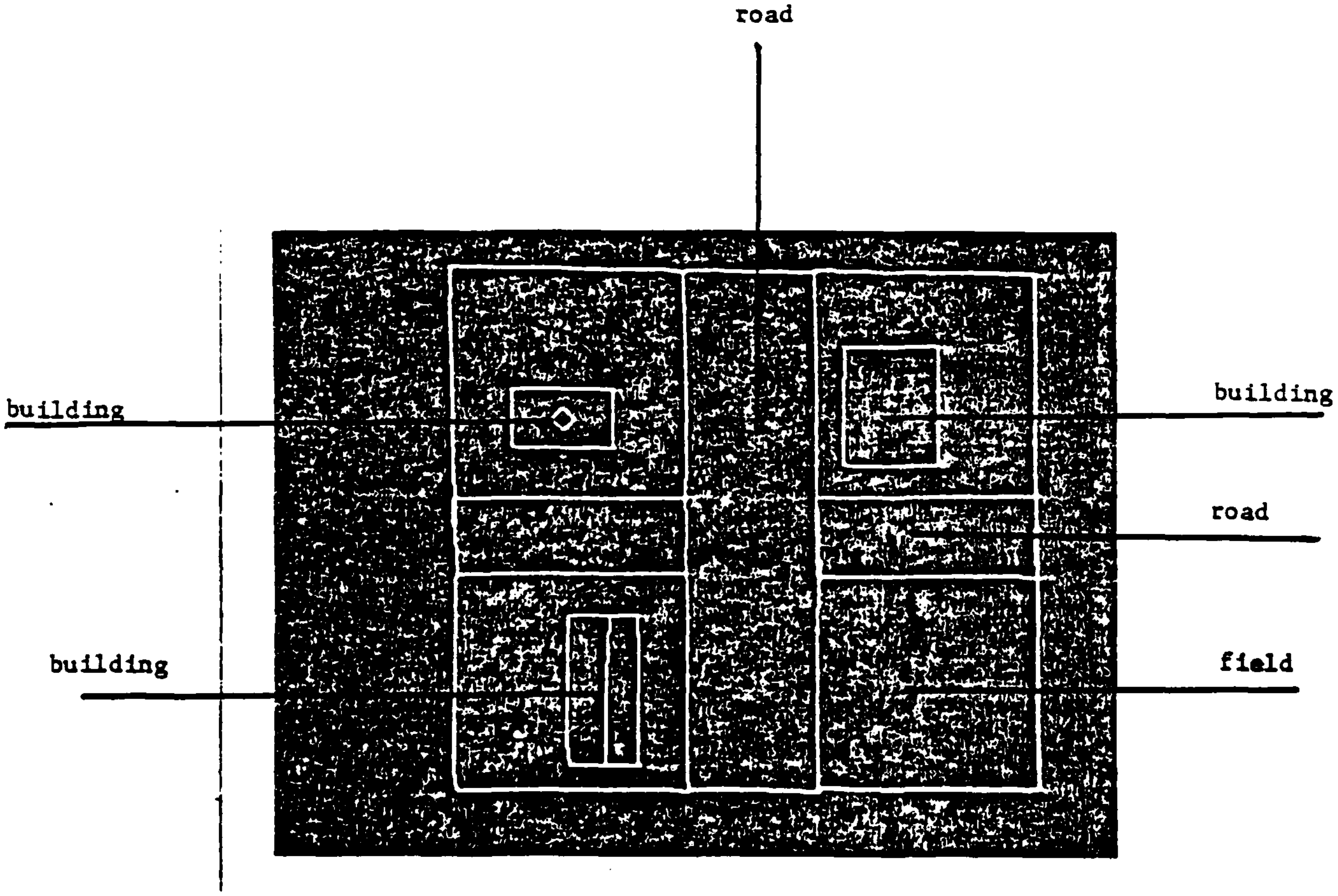


Figure 8-10: Surface graph data structure.

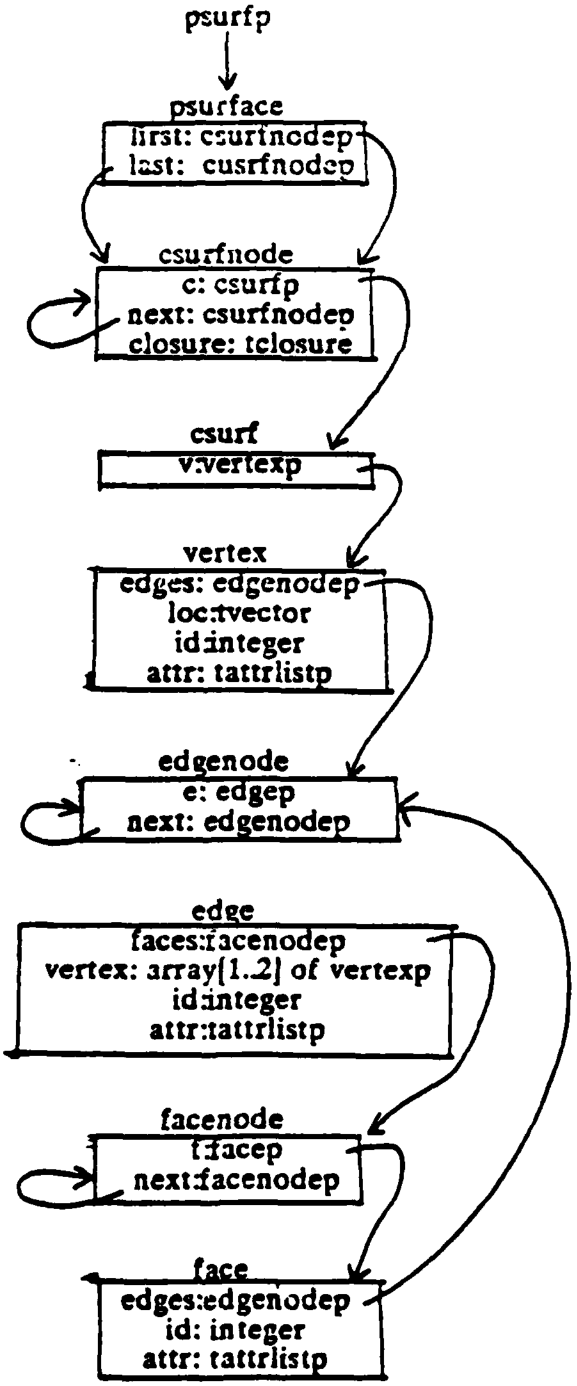


Figure 8-11: Reconstructed planar surfaces, rendered by Movie.BYU in top view.